Organization-Oriented Systems: Theory and Practice

Gil Tidhar

February, 1999

Submitted in total fulfilment of the requirements of the degree of Doctor of Philosophy

Department of Computer Science and Software Engineering
The University of Melbourne
AUSTRALIA

© Copyright Gil Tidhar, 1999
It is easier to pull large things apart, breaking them into smaller pieces, than to put small pieces together into something sensible.
Abstract

We investigate the problem of developing a formal language for specifying and reasoning about real-time embedded distributed computer systems. In particular we investigate the problem of developing a theoretical framework for specifying and analyzing different aspects of real-time embedded distributed coordination. In addition to the theoretical framework we also consider the practical aspects of developing real-time embedded distributed systems.

The main contributions are: theoretical models for specifying real-time embedded distributed systems; a new approach to the specification of coordination in real-time embedded distributed systems; and a new methodology and development environment for implementing such coordination techniques.

We present a hierarchy of models of real-time embedded distributed systems with increasing levels of expressiveness. At the top of the hierarchy is the Organization-Oriented Model. This model is inspired by Organization and Management Theory. It has organizations as primitives and explicit representation of joint mental attitudes, social mental attitudes, and team structure. We use the Organization-Oriented Model as the basis for the investigation of different types of coordination and decision making techniques. To highlight the differences between the various models in the hierarchy we use a running example from the Air Mission Modelling domain.

We address the practical aspects of developing real-time embedded distributed systems by presenting an Organization-Oriented Development Environment. We also present a preliminary analysis and design methodology for developing organization-oriented systems. The methodology and development environment have been used to develop a prototype organization-oriented system for modelling air missions.
Acknowledgements

I would first and foremost like to thank Ravit, my friend and wife, for her love, support, and understanding. I am sure that without her efforts in keeping our home functioning I would not have been able to complete this work (even if it is just because the electricity, phone, and gas would have been disconnected long ago). I would like to thank her for giving me advice when I needed it and just being there for me without question when I needed that. In particular for not asking the question: “why are you doing this and is it worth all this effort?”

Very close after Ravit in the contribution to this work come my three sons: Ron, Noam, and lately also Tom. I would especially like to thank them for how extraordinarily forgiving and understanding they were. I can not count the number of times I was working at home and although I was there in flesh (i.e., “the lights were on”) my mind was in a place where organizational structures collide with possible worlds (i.e., “there was nobody home”).

I now understand that the role of a supervisor of a PhD student is not very different from the role of a parent of a three year old child. Both have to encourage, direct, and discipline. All this while still giving the student (or child) the freedom to explore their own ideas and suffer the consequences of making the wrong decisions. I would like to thank both Liz Sonenberg and Anand Rao, my supervisors, for their patience and understanding. In particular their patience for my (at times unreasonable) strong views. I was amazed how many times they either complemented each other in their comments or were simply consistent with each other without coordinating their views in advance. They clearly demonstrate the saying: “Great minds think alike.”

The preliminary part of this work was completed while I was working at the Australian Artificial Intelligence Institute (AAII) and the final part while I was working at the University of Melbourne. I would like to thank Leon Sterling, Head of Computer Science at the University of Melbourne, for his support, encouragement, understanding, and constructive comments. I would like to thank the various researchers at AAII for the opportunity to interact with them.

During this period I also had the good fortune of cooperating with Lawrence Cave-don from the Royal Melbourne Institute of Technology (RMIT). I would like to thank him for the opportunity to work together and for demonstrating to me that philosophy, mathematics, cognitive science, and a good sense of humour can all mix quite naturally.

I would like to express my gratitude to Clint Heinz, Ian Lloyd, and Brad Smith from Air Operations Division, Australian Defence Science and Technology Organization (DSTO), for providing me with the operational examples and domain knowledge for the Air Mission Modelling. These examples and domain have been used by myself as a form of “reality check” for the ideas presented in this work. I would particularly like to thank Dino Appla from Air Operations Division, DSTO, for championing the concept of multi-agent systems within his organization. I found his enthusiasm quite inspiring.

During my work on the thesis there were a number of individuals that assisted me in different aspects of this work. I would like to thank Clint Heinz from the Air Operations Division, DSTO for the interesting discussions we had and his insight and valuable
comments on the methodology for developing Organization-Oriented Systems (described in Chapter 12). I would also like to thank Paolo Busetta and Andrew Hodgson, now both at Agent Oriented Software, for their assistance in understanding the dMARSTM compiler and interpreter. Without their support I would not have been able to develop the TOP system (described in Chapter 12). Tim Gabric and Simon Ch’ng, then from the University of Melbourne, provided valuable comments on the theory and implemented the ideas on Guided Team Selection (described in Chapter 11). I would also like to thank Milind Tambe from the Information Sciences Institute (ISI) for the interesting discussions we had about teams and team plans.

There were many other people who have listened, advised, encouraged, and forgiven over the past few years. They include, amongst many others, Jim Butler, Andrew Lucas, Simon Goss, Graeme Murray, the members of the Agents Lab at the University of Melbourne, Van Parunak, Ralph Ronnquist, Steffan Bussman, and last but not least my sister Ronit, my parents Galia and Shlomo Tidhar, and my parents-in-law Rocha and Yoel Gruner.
Contents

1 Introduction 1
   1.1 The Problem and Main Contribution: Summary 5
   1.2 Assumptions 7
      1.2.1 Problem Characteristics 7
      1.2.2 System Design 8
      1.2.3 Component Characteristics 8
   1.3 Models of Agents, Multi-Agents, and Organizations 10
      1.3.1 Single Agent Systems 10
      1.3.2 Multi-Agent Systems 11
      1.3.3 Human Organizations 11
   1.4 An Overview of the Thesis 14

2 Example Domain 19
   2.1 Air Missions 19
   2.2 Air Mission Scenario 20
      2.2.1 The Operational Scenario 21
      2.2.2 Analysis 23

3 Required Features and Relevant Approaches 27
   3.1 Requirements 27
      3.1.1 Interactions Between a System and its Environment 29
      3.1.2 Representing Organizations 29
      3.1.3 Organization Formation and Dismantling 31
      3.1.4 Organization Decision Making 32
      3.1.5 Payoffs, Costs, and Probabilities 34
   3.2 Related Approaches 35
      3.2.1 Models of Organizational Structures 35
      3.2.2 Formal Organizational Models 36
      3.2.3 Command, Control, and Communication 43
      3.2.4 Bounded Rationality 45
      3.2.5 Philosophical Models 47
      3.2.6 Formal Mathematical Models 48
      3.2.7 Approaches to Developing Multi-Agent Systems 49
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.8</td>
<td>Other Related Approaches</td>
<td>52</td>
</tr>
<tr>
<td>3.3</td>
<td>Limitations of Related Approaches</td>
<td>54</td>
</tr>
<tr>
<td>4</td>
<td>Single Agent Systems</td>
<td>57</td>
</tr>
<tr>
<td>4.1</td>
<td>Syntax of a Single Agent BGI System</td>
<td>58</td>
</tr>
<tr>
<td>4.2</td>
<td>Modelling Air Missions with a Single-Agent System</td>
<td>59</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Plans and Coordination</td>
<td>60</td>
</tr>
<tr>
<td>4.3</td>
<td>Possible-Worlds Semantics</td>
<td>62</td>
</tr>
<tr>
<td>4.4</td>
<td>Basic Axiom System</td>
<td>64</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Axiomatization of CTL Component</td>
<td>64</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Axiomatization of BGI component</td>
<td>65</td>
</tr>
<tr>
<td>4.5</td>
<td>Soundness and Completeness</td>
<td>66</td>
</tr>
<tr>
<td>4.6</td>
<td>Small Model Theorem</td>
<td>67</td>
</tr>
<tr>
<td>4.7</td>
<td>Algorithm</td>
<td>71</td>
</tr>
<tr>
<td>4.8</td>
<td>Completeness</td>
<td>74</td>
</tr>
<tr>
<td>5</td>
<td>Multi-Agent Systems</td>
<td>76</td>
</tr>
<tr>
<td>5.1</td>
<td>Syntax</td>
<td>76</td>
</tr>
<tr>
<td>5.2</td>
<td>Possible-Worlds Semantics for Multi-Agent Systems: Informal Discussion</td>
<td>77</td>
</tr>
<tr>
<td>5.2.1</td>
<td>The Domain</td>
<td>77</td>
</tr>
<tr>
<td>5.2.2</td>
<td>A Model of Events</td>
<td>77</td>
</tr>
<tr>
<td>5.2.3</td>
<td>A Model of Time</td>
<td>78</td>
</tr>
<tr>
<td>5.2.4</td>
<td>A Model of an Agent</td>
<td>78</td>
</tr>
<tr>
<td>5.3</td>
<td>Modelling Air Missions with a Multi-Agent System</td>
<td>79</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Mental State</td>
<td>79</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Plans and Coordination</td>
<td>80</td>
</tr>
<tr>
<td>5.4</td>
<td>The Formal Semantic Model of Multi-Agent Systems</td>
<td>81</td>
</tr>
<tr>
<td>5.5</td>
<td>Axioms</td>
<td>82</td>
</tr>
<tr>
<td>5.6</td>
<td>Soundness and Completeness</td>
<td>82</td>
</tr>
<tr>
<td>6</td>
<td>Team-Oriented Systems</td>
<td>86</td>
</tr>
<tr>
<td>6.1</td>
<td>Possible Worlds Semantics for Team-Oriented Systems: Informal Discussion</td>
<td>87</td>
</tr>
<tr>
<td>6.1.1</td>
<td>A Model of a Team</td>
<td>88</td>
</tr>
<tr>
<td>6.2</td>
<td>Syntax</td>
<td>92</td>
</tr>
<tr>
<td>6.3</td>
<td>Modelling Air Missions with a Team-Oriented System</td>
<td>93</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Teams</td>
<td>93</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Joint Mental Attitudes</td>
<td>94</td>
</tr>
<tr>
<td>6.3.3</td>
<td>Team Plans and Coordination</td>
<td>95</td>
</tr>
<tr>
<td>6.4</td>
<td>The Formal Semantic Model of Team-Oriented Systems</td>
<td>95</td>
</tr>
<tr>
<td>6.4.1</td>
<td>The Model</td>
<td>95</td>
</tr>
<tr>
<td>6.4.2</td>
<td>Syntactic and Semantic Conditions</td>
<td>98</td>
</tr>
<tr>
<td>6.5</td>
<td>Soundness and Completeness</td>
<td>99</td>
</tr>
</tbody>
</table>
CONTENTS

7 Axiom Systems for Team-Oriented Systems 105
  7.1 Mutual Beliefs, Joint Goals, and Joint Intentions for Every Sub-team . . . 106
    7.1.1 Syntactic Conditions (Axioms) . . . . . . . . . . . . . . . . . . . . 106
    7.1.2 Semantic Conditions . . . . . . . . . . . . . . . . . . . . . . . . . . 107
    7.1.3 Soundness and Completeness . . . . . . . . . . . . . . . . . . . . . 108
    7.1.4 Comparing BGI_{\text{CTL}}^{\text{os-e}} and Halpern and Moses’s \text{E}_G . . . . . . . . . 112
  7.2 Joint Mental Attitudes as Common Modalities (or Self Referential Axioms) 113
    7.2.1 Syntactic Conditions (Axioms) . . . . . . . . . . . . . . . . . . . . 114
    7.2.2 Semantic Conditions . . . . . . . . . . . . . . . . . . . . . . . . . . 114
    7.2.3 Soundness and Completeness . . . . . . . . . . . . . . . . . . . . . 115
    7.2.4 Comparing BGI_{\text{CTL}}^{\text{os-c}} and Halpern and Moses’s \text{C}_G . . . . . . . . . 120
  7.3 Introspection in a Team-Oriented Model . . . . . . . . . . . . . . . . . . . 121
  7.4 Introspection in the Presence of the Every Sub-team Axioms . . . . . . . . 122
  7.5 Team-Oriented Systems and Bounded Rationality . . . . . . . . . . . . . . 124
    7.5.1 Bounded Rationality in Organizational Theory . . . . . . . . . . . . 124
    7.5.2 The Complexity of Validating a Mutual Belief . . . . . . . . . . . . 126
    7.5.3 Bounded Rationality in Team-Oriented Systems . . . . . . . . . . . . 127

8 Organization-Oriented Systems 129
  8.1 Organization-Oriented Systems: Informal Discussion . . . . . . . . . . . . 130
    8.1.1 Background and Motivation . . . . . . . . . . . . . . . . . . . . . . 130
    8.1.2 Informal Discussion on The Formal Model . . . . . . . . . . . . . . 132
    8.1.3 Rationality in Organizations . . . . . . . . . . . . . . . . . . . . . 134
  8.2 Syntax . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 135
  8.3 Air Missions and Organization-Oriented System . . . . . . . . . . . . . . . 136
    8.3.1 Organizations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 136
    8.3.2 Organizational Structures . . . . . . . . . . . . . . . . . . . . . . . 138
    8.3.3 Organizational Plans and C3 Structures . . . . . . . . . . . . . . . 140
  8.4 Possible Worlds Semantics . . . . . . . . . . . . . . . . . . . . . . . . . . 144
  8.5 Soundness and Completeness . . . . . . . . . . . . . . . . . . . . . . . . 147
  8.6 The Spectrum of Social and Joint Mental Attitudes . . . . . . . . . . . . . . 151
  8.7 Interactions Between Different Social Mental Attitudes . . . . . . . . . . . . 154
  8.8 Interaction Between Social Mental Attitudes and Joint Mental Attitudes . . . 155
    8.8.1 Basic Organization-Oriented System . . . . . . . . . . . . . . . . . 156
    8.8.2 Soundness and Completeness . . . . . . . . . . . . . . . . . . . . . 158
  8.9 Interaction Between Social Mental Attitudes and Team Structure . . . . . . 164
  8.10 A Model of Organizational Structures . . . . . . . . . . . . . . . . . . . . 164

9 Axiom Systems for Organization-Oriented Systems 166
  9.1 The Dominant Coalition . . . . . . . . . . . . . . . . . . . . . . . . . . . 167
  9.2 A Military Unit . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 168
    9.2.1 Transitivity of the Social Mental Attitudes . . . . . . . . . . . . . . 168
    9.2.2 Sub-team Transitivity of the Social Mental Attitudes . . . . . . . . . 169
9.2.3 Constraints on Control and Communication .................................. 170
9.3 The Simple Organization ................................................................. 170
9.4 The Divisionalized Organization ......................................................... 171

10 Operational Semantics 174
10.1 Operational Semantics for Agents, Teams, and Organizations .............. 174
  10.1.1 Operational Semantics for Agent-Oriented Systems ..................... 175
  10.1.2 Operational Semantics for an Agent in a Multi-Agent Systems ...... 177
  10.1.3 Operational Semantics for Team-Oriented Systems ..................... 180
  10.1.4 Operational Semantics of Organization-Oriented Systems .......... 182
10.2 The Specifications ................................................................. 183
  10.2.1 Social Structures .............................................................. 183
  10.2.2 Organizational Plans .......................................................... 186
  10.2.3 Internal Organizational Model .............................................. 188
10.3 An Architecture for Organizations and Organizational Behavior .......... 189
  10.3.1 The Components .............................................................. 189
  10.3.2 Informal Description .......................................................... 190
  10.3.3 The Processes ................................................................. 195
10.4 Coordinating Multiple Organizations ............................................ 200
10.5 Evaluation of the Organization-Oriented Approach ........................... 201
  10.5.1 Usability ................................................................. 202
  10.5.2 Complexity ................................................................. 204
  10.5.3 Further Evaluation .......................................................... 205

11 Guided Team Selection 208
11.1 Introduction ................................................................. 208
11.2 Roles and Allocations ............................................................ 210
  11.2.1 Role Specification .......................................................... 211
  11.2.2 Allocation Specification ..................................................... 212
11.3 Team Selection ................................................................. 213
  11.3.1 Relevant Teams ............................................................. 214
  11.3.2 Applicable Teams ............................................................ 215
11.4 Analysis ................................................................. 216
  11.4.1 Formulation of Processes .................................................... 216
  11.4.2 Effects on Means-End Tree ................................................ 218
11.5 Comparison with Relevant Work ................................................ 220
11.6 Conclusions ................................................................. 221

12 System Design 223
12.1 Constraints and Assumptions ....................................................... 223
12.2 High Level Design .............................................................. 224
  12.2.1 Organizational Structures .................................................. 225
  12.2.2 Social Awareness ............................................................ 227
CONTENTS

12.2.3 Organization Formation ........................................ 229
12.2.4 Plan Language .................................................. 229
12.2.5 Operational Semantics of Plans ................................. 231
12.2.6 The Agent’s Control Mechanism ............................... 231
12.2.7 Organization-Oriented Development Environment .......... 234
12.3 Detailed Design .................................................... 234
  12.3.1 Default Event Handlers ...................................... 237
  12.3.2 Transformer .................................................. 238
12.4 Analysis and Design Methodology for Organization-Oriented Systems .................................. 239
  12.4.1 The Components of an Organization-Oriented System ... 240
  12.4.2 The Methodology ............................................ 240

13 Conclusions .......................................................... 247
  13.1 Review of Thesis .................................................. 248
  13.2 Possible Extensions and Further Investigations ............... 250
  13.3 Main Contribution .............................................. 252

A Default Event Handlers .............................................. 266
# List of Figures

2.1 An operational air mission scenario. ........................................ 21

5.1 An example of a pseudo-BGI$^{\text{mas}}_{\text{CTL}}$-tableau. ................................. 84

6.1 An example of a pseudo-BGI$^{\text{tos}}_{\text{CTL}}$-tableau. ................................. 99

6.2 An example of a pseudo-BGI$^{\text{tos}}_{\text{CTL}}$-tableau. ................................. 102

7.1 An example of a pseudo-BGI$^{\text{tos}-e}_{\text{CTL}}$-tableau. ................................. 110

8.1 A specification of the FC organization family. ................................. 137

8.2 A specification of a tactical control organizational structure. .................. 139

8.3 A specification of a communication relation. ................................. 140

8.4 A specification of a control relation. ................................. 141

8.5 An organizational plan for performing a pincer intercept. .................. 143

8.6 An example of a pseudo-BGI$^{\text{tos}}_{\text{CTL}}$-tableau. ................................. 149

8.7 An example of a pseudo-BGI$^{\text{tos}-b}_{\text{CTL}}$-tableau. ................................. 161

10.1 Control Loop of an Agent in a Single-Agent world. .................. 177

10.2 Control Loop of an Agent in a Multi-Agent world. .................. 179

10.3 Control Loop of a Team. ................................. 181

10.4 Control Loop of an Organization. ................................. 192

11.1 Operational semantics of reactive planning and guided team selection. .. 213

11.2 Average relevant teams per goal (with $l = 2, q = 1, m = 2$). .......... 219
Chapter 1

Introduction

It is hard to imagine modern life without computer systems - they have simply become an integral part of our day to day experiences. They are used in controlling the equipment that we use (e.g., television, air-conditioning, telephone, etc.), providing us with services (e.g., automatic teller machines, vending machines, personal computers, etc.), and assisting others in providing us with these services (e.g., banks, supermarkets, motor mechanics, etc.). Such computer systems typically operate as a background activity and are hidden from users.

Specialization of computer systems is the basis for the enormous value that they bring. Each system incorporates substantial information and knowledge about the state of the device it is embedded in and the ways to control it. The design of these systems would typically incorporate the substantial knowledge and expertise used by the software, electronics, and mechanical engineers that built them.

In a dynamic world, the environment changes while the computer system is achieving its objective. Such changes may be caused by other computer systems that operate in the same environment, e.g., the hot water system starting to consume electricity required by the air-conditioning system, or just because that is the way the world operates, e.g., the temperature outside drops towards nightfall.

Not all computer systems are designed to consider the dynamic aspects of the world. Even if they do identify the change in the environment they still continue with their current operation ignoring that change. A system that interacts with its environment and adjusts its behavior in accordance with the changes to the environment is referred to as an embedded system.

The importance of embedded systems comes from the requirement that the computer system should be able to achieve its objectives despite the changes to the environment (one could refer to such a system as a survivable system). Furthermore the computer system should ensure that despite the changes to the environment in which it operates it still achieves the best possible result.

It is clear that developing a computer system that will adjust its behavior to any possible change in the environment is not a simple task. Even if we restrict the computer system to adjust to limited types of changes, the task is not simple. This is because the
same change may occur at different times.

Although a computer system may adjust its behavior in response to the changing world, the speed with which the world changes and the speed with which the system responds may be critical to its safe and efficient operation. A computer system that reacts with a speed which is comparable to the speed with which the world changes is referred to as a real-time system.

Again it is clear that developing a computer system that will react at any possible speed is not a simple task. Even if we restrict the computer system to react no slower than a particular speed it is still difficult to guarantee that the actual reaction will be the right (or best) one.

The above behavior should be exhibited in the context of the resources available to computer systems. Such resources include processors used for performing computations, memory used to store information, communication networks to interact with other computers, and time used to perform their tasks. Like other physical systems computer systems also have limited resources.

Artificial Intelligence is a field of Computer Science that focuses on developing advanced computer science techniques. These techniques are used for building computer systems that provide functionality that can not be provided using conventional computer science techniques. Researchers in Artificial Intelligence typically look to other disciplines for inspiration. Such disciplines include economics, social sciences, philosophy, mathematics, biology and many others.

An Intelligent Software Agent is a type of computer system that uses artificial intelligence techniques. We refer to an intelligent software agent in short as an agent. There are many definitions of an agent. The definition adopted here (c.f. Wooldridge and Jennings [152]) is that:

"an agent is an embedded real-time computer system that also includes the following characteristics:

**Autonomy:** the ability to operate without the direct intervention of humans or others, and have some kind of control over its actions and internal state;

**Pro-activeness:** the ability to not simply act in response to the changing environment, but also to exhibit goal-directed behavior by taking the initiative."

There are different approaches to the representation of an agent and its behavior. We refer to a particular approach as a model of an agent. We would say that there are different

---

1It is the predicament of researchers in Artificial Intelligence that whenever an advanced computer science technique they developed becomes widely used it is referred to as a conventional technique.

2Wooldridge and Jennings also include social ability. That is, the ability to interact with other agents (and possibly humans) via some kind of agent-communication language. We consider this ability relevant only if the environment of the agent includes other agents that it can communicate with. We will discuss this issue below when discussing multi-agent systems.
models and would analyze their strengths and limitations. One should note that each model will also determine the ease in which a developer can develop the computer system.

We have discussed the development of a single embedded real-time computer system. As mentioned above our modern environment includes a variety of computer systems that control different equipment, provide us with a service, and assist others to provide us with a service.

Networking of many computer systems is now a reality. Such networking can either be done using specialized networks (e.g., a computer network in a bank), the telephone network (e.g., telephone companies billing and call-waiting services, tele-computing, etc.), and even using the power network (e.g., monitoring of electricity consumption).

The networking of computer systems provides the connectivity between them. Such connectivity allows them to exchange information. Nevertheless, different computer systems that operate in the same environment do so in parallel and in many cases independently of each other (e.g., the television may be on while you are speaking on the telephone).

Like isolated specialists, the specialized computer systems are typically ignorant of the other computer systems that operate in the same environment. They simply are not designed to have information about other systems or to exchange information with systems that operate in the same environment.

This can lead to situations in which we as users are inconvenienced by this environment or situations in which two systems degrade the performance of each other. Examples of such situations include: the television interfering with a telephone conversation, the telephone ringing in the middle of the most dramatic moment of the film, the air-conditioner and hot water system working in full capacity and causing the electricity fuse to blow, etc.

Cooperation is the key to the solution of many of these problems. That is, exactly like with human specialists we need to make the computer systems work together towards a common objective. The benefits are gained by coordination of the activities in an attempt to minimize the potential for conflict and wastage in the overall system.

**Coordination** is defined according to the Webster dictionary as the harmonious functioning of parts for effective results and **to coordinate** is defined as to bring into a common action, movement, or condition.

Indeed there are already some computer systems that coordinate their activities to provide better facilities for their users. Such systems include amongst others: (1) a security system that monitors the location of individuals and initiates the appropriate call diversion; (2) an audio-visual presentation system that interacts with the lighting controls in a lecture theatre; and (3) a hot-water system that consumes electricity during low cost hours as determined by the electricity meter.

Assigning responsibilities and identifying interactions to each of the components of the system are part of improving the cooperation and performance of the system. Responsibilities are relevant both with respect to the specialization of each system (e.g., which system controls the hot water system) and overall decisions that have to be taken. The interactions identify:
1. which component should convey what information to what other component (e.g.,
does the hot water system need to know what the current electricity price is, does it
need to know if the telephone is ringing, etc.);

2. which component can instruct another component to achieve an objective (e.g., when
a movie is on and the phone rings, can the telephone instruct the television to lower
its volume, alternatively should the television instruct the telephone to divert all calls
to the answering machine while the movie is on, etc.); and

3. which component determines what is the best way of achieving an objective (e.g.,
should the electricity meter allocate time to the hot water system and the air-
conditioning, should the air-conditioning and hot water systems independently try to
minimize their consumption based on information from the electricity meter, etc.).

A Distributed Computer System is a set of computer systems that operate in parallel
on equipment that is connected together. We consider distributed systems in which the
overall system, including all its components, is designed to achieve common objectives. The
components have different types of responsibilities and interactions and use coordination
to improve the overall system performance.

Examples of common objectives include: controlling of the home or office environment,
controlling the flow of traffic in a city, controlling the utilization of telephone exchanges,
or achieving a complex financial transaction. One can view cooperation, coordination,
responsibilities, and interactions as one type of means that the distributed system employs
to achieve its objectives. We refer to these means as the distributed behavior.

The allocation of responsibilities and the interactions between the components of the
distributed system provides for an order in the complex distributed behavior. This order
specifies the way the system is “organized”. We refer to this as the structure of the
distributed computer system.

Like a single computer system a distributed computer system can also be required to
respond to the changing environment and to do so under real-time constraints. Such a
distributed computer system is referred to as a real-time embedded distributed system.

The problem of developing a real-time embedded distributed system is much more
complex than that of developing a single real-time embedded computer system. This is
because one also has to consider the cooperation and coordination aspects. Furthermore,
one may not be able to assume that a single cooperation and coordination mechanism
would be suitable under all circumstances.

The importance of adjusting the distributed behavior comes from the requirement
that the distributed computer system should be able to complete its objective despite the
changes to the environment (one could refer to such a system as a survivable distributed
system). Furthermore, the distributed computer system should ensure that despite the
changes to the environment it still achieves its objectives in the best possible way.

As in the case of a single computer system, it is clear that developing a distributed
computer system that will adjust its distributed behavior to any possible change in the
environment is not a simple task. Even if we restrict the distributed computer system to
adjust to limited types of changes, the task is not simple since the same change may occur at different times.

A distributed computer system may adjust its behavior in response to the changing world. Nevertheless, the speed with which the world changes and the speed with which the distributed system responds may be critical to a safe and efficient operation.

A multi-agent system is a real-time embedded distributed system. As a stream of artificial intelligence, multi-agent systems would include advanced techniques for representing the distributed behavior of the distributed computer system. That is, the techniques used for cooperation, coordination, assignment of responsibilities, and identification of interactions between the components of the system. In the context of a multi-agent system we refer to the distributed behavior as the social behavior and to the structure of the distributed system as the social structure.

Note that some multi-agent systems implement social behavior and structure using communication. For such systems one should add the additional characteristics to the agent as described by Wooldridge and Jennings [152]. That is:

“social ability: the ability to interact with other agents (and possibly humans) via some kind of agent-communication language.”

Again there are different approaches to the representation of a multi-agent system. We would refer to each approach as a model of a multi-agent system. We can compare the advantages and disadvantages of each model. Again this will include the ease in which the multi-agent system can be developed using the model. We refer to such a system as an Organization-Oriented System.

1.1 The Problem and Main Contribution: Summary

It is hard to imagine modern life without computer systems - they have simply become an integral part of our day to day experiences. They control the equipment that we use and the environment in which we live, they provide us with services, and they assist others in doing so. In many cases such computer systems operate under real-time and resource constraints. Furthermore, they perform operations which are at times safety critical. Many of these computer systems operate in a dynamic environment. Given that they are embedded in the same environment the actions they take affect each other. More importantly these actions affect the users that operate in this environment.

With increased networking the connectivity between such computer systems is increasing. Nevertheless, most computer systems do not coordinate their activities. Furthermore, they do not make decisions that improve the overall performance of the system and the services provided to the user. They are like experts that operate in isolation.

One can find analogies to the development of computer systems in the development of human society. Current computer systems can be viewed as being in early stages of social development: each individual is relatively advanced in its survival skills, but language is still in its infancy and cooperation between individuals is very basic. It is obvious that
computers are not humans and vice versa. Nevertheless, one could be inspired by models of human coordination. Indeed there is a field of social science, known as Organization and Management Theory (OMT), that investigates coordination techniques and social structures in human organizations. Work done in OMT may be used in developing a theory of real-time embedded distributed coordination for computer systems.

In this thesis we investigate the problem of developing a language for specifying real-time embedded distributed systems. We focus on the specification of coordination and distributed decision making techniques. There may be many ways to define such techniques. We investigate the problem of developing frameworks for defining and analyzing different aspects of real-time embedded distributed coordination.

To be able to precisely specify the desired system characteristics one has to use “precise languages”, i.e., formal languages. We are particularly interested in formal frameworks in which both the desired system characteristics and the specification of this characteristics are formally defined. That is, we require formal semantic and axiomatic models of a real-time embedded distributed system. Note that, in order to enable reasoning interchangeably between the semantic and axiomatic models we need that they are sound and complete with respect to each other.

In addition to a theoretical framework we also consider the practical aspects of developing real-time embedded distributed systems. In particular the development of an analysis and design methodology and a corresponding development environment. We focus on a model and methodology that support a form of abstraction suitable for a process of “Top-Down” specification of distributed systems.

In the following chapters we provide a hierarchy of models of real-time embedded distributed systems. The models presented differ in their expressiveness. In the simpler models only the most basic components of the distributed system can be specified. Furthermore the coordination of these components is implicit in the specification and can not be easily modified.

At the top of the hierarchy we provide the Organization-Oriented Model. Inspired by ideas from OMT, this model allows the developer to explicitly specify at a higher level of abstraction the components of the distributed system. Furthermore, the developer can explicitly specify the coordination of these components. We show how the more expressive models generalize some aspects of the simpler models.

We use the organization-oriented model to investigate different aspects of real-time embedded distributed systems. In particular we focus on the possible relationships between the components of the system and the system as a whole. We demonstrate how some human organizational structures and coordination mechanisms can be specified using these models.

For each of the models in the hierarchy we provide both a semantic and axiomatic model. We prove the soundness and completeness of these models with respect to each other. We also provide the operational semantics for the languages provided. We particularly focus on the operational semantics for an Organization-Oriented System.
In addition to the hierarchy of models we use a practical domain, the Air Mission Modelling domain, as a running example throughout the thesis. We demonstrate how an air mission modelling system would be developed using each of the models in the hierarchy.

To address the practical aspects of developing such systems we have designed and implemented a prototype Organization-Oriented Development Environment. This development environment includes graphical editors, compilers, and communication mechanisms. To facilitate the development process we provide a preliminary Organization-Oriented Analysis and Design Methodology. The methodology and development environment have been used to develop a prototype organization-oriented system for modelling air missions.

1.2 Assumptions

In previous sections we have made some implicit assumptions about the characteristics of the distributed system and its required behavior. We would now like to make these assumptions explicit, and describe how these assumptions affect this work. We identify the following types of assumptions:

1. **Problem Characteristics**: These are assumptions regarding the general characteristics of the problems that the organization-oriented system is required to address. Such assumptions include a requirement for real-time performance, an assumption that communication facilities are reliable, etc.

2. **System Design**: These are assumptions regarding the system design process adopted when building the organization-oriented system. Such assumptions include centralized design, distributed system, etc.

3. **Component Characteristics**: These are assumptions regarding the characteristics of the components that are part of the system developed. Such assumptions include multiple goals, heterogeneous ability, etc.

1.2.1 Problem Characteristics

The development of agent-oriented and organization-oriented systems has emerged to provide an answer to the requirements put forward by a particular class of problems. The assumptions made with respect to the characteristics of the class of problems to be solved include:

1. **Dynamic**: The domain is dynamic and may change while the system is responding to previous changes or while it is achieving goals.

2. **Real-Time**: Changes in the environment are frequent and require real-time response of the system.
3. **Distributed:** The problem is naturally distributed either geographically or functionally. That is, information is available in separate locations or there are distinct functionality which is required from the system which should be performed simultaneously.

4. **Connectivity:** Reliable distributed communication facilities are available and are not part of the problem. Distributed components are able to communicate with other components either directly or indirectly. Indirect communication exists when a component can communicate with another component via a third component communicating on its behalf (directly or indirectly).

### 1.2.2 System Design

The design process and the characteristics of the system design will affect the suitability of different knowledge representation and the type of algorithms that can or should be used. The assumptions made with respect to the nature of the system design include:

1. **Centralized Design:** We assume that the design of the distributed system and the specification of the social behavior used by all the components is done in a centralized way. That is, it is either done by a central designer or it is done by multiple designers tightly coordinating their activities. The detailed design of each individual component can be done independently (assuming it complies with the overall system design). There are other researchers in multi-agent systems that do not make this assumption. They typically focus on ways of improving the emergence of social behavior in a bottom-up approach (see for example Drogoul [31]).

2. **Distributed:** The system is a distributed system composed of autonomous components.

3. **Embedded:** The system is embedded in the environment receiving changes as they happen and affecting the environment by acting.

4. **Knowledge Acquisition:** The expected behavior of the system is well defined and the knowledge required to solve the problem is known in advance.

5. **Reliability:** The behavior of each component and the behavior of the system as a whole is predictable and verifiable.

### 1.2.3 Component Characteristics

As we are dealing with software components we are able to make assumptions regarding the behavior, internal processes, and knowledge available to them. We assume that the components have the following characteristics:
1.2. ASSUMPTIONS

1. **Component Rationality**: The definition of rationality adopted here follows from the work of Rao and Georgeff [87, 90]. That is, it is a relationship between the beliefs, goals, and intentions held by the component.\(^3\) Here we refer to this type of relationship as *Component Rationality*. The designer of a component may adopt any type of component rationality.

2. **Multiple Goals**: At any given time the components may possess more than one goal. For each such goal there would be a single intention to achieve this goal. A component may possess multiple intentions.

3. **Common Goals**: We assume that the components of the system share common goals. These goals represent the objective of the system as a whole. That is, they represent various aspects of the problem as described above. There are other researchers in multi-agent systems that do not make this assumption. They typically focus on mechanisms in which individual agents can improve their individual objectives in the context of other agents (see for example Zlotkin and Rosenschien [94]).

4. **Social Rationality**: As components have both individual and joint mental attitudes there is some relationship between these two types of mental attitudes.\(^4\) The relationships between the individual, joint mental attitudes, and social structure are referred to as *Social Rationality*. The designer of the components may adopt any type of social rationality.

5. **Heterogeneous Ability**: The components have varying abilities some of which may be overlapping. The notion of ability adopted here has several different meanings and we will note explicitly which of them is used when. The notion of ability will be used to denote: (1) ability to achieve a goal; (2) ability to perform a plan; and (3) ability to play a role within a team. These notions correspond to similar concepts described by us previously [124].

6. **Organizational Ability**: The knowledge about the relationship with other components and the ability to reason about this knowledge exists in all components.

7. **Bounded Ability**: The components have limited abilities. These limitations are with respect to both the types of actions they can take as well as the resources available to them (computational and data storage resources). The characteristics of the problem are such that they can not solve the problem individually but rather they are required to cooperate.

8. **Communication**: Components are able to send and receive structured messages to and from other components.

\(^3\)For example, a component adopting *Strong Realism* can only adopt a goal to achieve a state of the world if it also believes that it can achieve it. Furthermore, a component can only intend to achieve a state of the world only if it has a goal to achieve it.

\(^4\)For example, components that give preference to their individual goals could be referred to as *selfish*. Components that give preference to the joint goals could be referred to as *benevolent*. 

© Copyright Gil Tidhar, 1999
1.3 Models of Agents, Multi-Agents, and Organizations

1.3.1 Single Agent Systems

A Belief-Goal-Intention (BGI) model of an agent is one particular approach to the specification and design of an agent. This model, also known as the Belief-Desire-Intention (BDI) model, has been formally described by Rao and Georgeff [88]. In this model an agent is comprised of information about its current state and the state of its environment (referred to as Beliefs), objectives it should achieve (referred to as Goals), and a course of action it has adopted in attempting to achieve its objectives (referred to as Intentions). The beliefs, goals, and intentions are referred to in general as the mental attitudes of the agent.

The behavior in a BGI model is based on the agent having multiple simultaneous goals. At any given moment there is only one goal that the agent would attempt to achieve - the most important one. Each goal would have its own intention. The agent can modify its beliefs, add or remove goals and intentions, and change the choice of the most important goal. After executing an action in the intention of the most important goal the agent will inspect its beliefs about the world and consider whether it should react to any changes.

The main advantage of a formal BGI model is that it is based on a formal mathematical theory with links to philosophy and decision theory. As such it allows the developer to verify and validate the design and implementation of the agent against the specification. This is a vital feature when considering mission critical computer systems.

The process of creating intentions is called planning. The naive approach to creating intentions is to re-evaluate all the possibilities every time the world changes. That is, compute all possible courses of actions that will achieve the objective, select the best course, and then perform the first action in this course. After this, check if the world has changed and if so repeat the same process. This approach is also referred to as planning from first principles.

The biggest problem with planning from first principles is that it is very inefficient and very time consuming. This would typically be unacceptable in real-time systems.

The pre-planned approach to creating intentions provides a way to overcome some of the problems of planning from first principles. In this approach an agent is provided with abstract specifications of possible courses of actions. These specifications are like “recipes” for achieving goals. Each such “recipe” is referred to as a plan. A plan is relevant only to a particular goal and is applicable under particular circumstances. A detailed account of the pre-planned approach has been provided by Rao [85].

In the pre-planned approach a plan includes a combination of actions to be performed and sub-goals to be achieved. Each sub-goal may be achieved using other plans. There may be multiple plans that are relevant to the same goal but are applicable under different circumstances.

In the pre-planned approach the process of creating an intention for a given goal is confined to: (1) the selection of a plan which is both relevant and applicable; and (2)
adoption of this plan as an intention. These simplifications markedly improve the response
time of the system. It is important to note that the choice of the plan to achieve a sub-goal
need only be done when that sub-goal has to be achieved. Given that the environment
is changing this allows the agent to select a plan only when necessary and select the best
plan under the most recent circumstances.

1.3.2 Multi-Agent Systems

Existing models of a multi-agent system have primarily focused on the development of
advanced agents that operate in an environment which has multiple agents. In particular
they focus on the basic social ability of the single agent.

Recall that the social behavior of the multi-agent system includes the cooperation and
coordination mechanisms. Also recall that the social structure of a distributed computer
system includes the assignment of responsibilities to the components and the identification
of the various interactions between them. Most models of multi-agent systems do not
represent the social behavior or structure explicitly. Rather, they are “hard-wired” into
the behavior of the individual agents.

In current models of multi-agent systems the developer of the system determines the
social behavior and structure that is required and then “translates” it into the required
individual behavior of each of the agents. The agents are then developed according to
these requirements. One can say that the social behavior and structure emerge from the
behavior of the individual agents in the multi-agent system. We refer to this approach as
the “bottom-up approach” to the development of multi-agent systems.

It is important to note that in the bottom-up approach there is no way to explicitly
specify the behavior of the distributed system as a whole or the behavior of any of its
sub-systems without explicitly specifying the behavior of the single agents.

BGI models of a multi-agent system have been recently proposed by a number of
researchers. They have primarily focused on adding to the BGI model of an agent an
abstract specification of the Beliefs, Goals, and Intentions of a set of agents. These are
referred to as Mutual Beliefs, Joint Goals, and Joint Intentions respectively. Examples
include the work of Kinny et al. [69] and Tambe [118].

The plans that are used by these sets of agents (referred to as Joint or Team Plans)
include a high-level specification of the coordination of the activities of the set of agents.
This provides the developer with a way of explicitly specifying the required coordination
for the set of agents. Furthermore it allows the set of agents to dynamically modify the
way it coordinates itself when attempting to achieve the same goal.

1.3.3 Human Organizations

Organization Theory has been focusing on investigating and developing models of human
organizations. In particular work has focused on developing explicit models of the structure
of the organization and the behavior of the individuals that are members of the organiza-
tion. These are referred to as organizational structures (see for example Mintzberg [79]) and organizational behavior (see for example Simon [104]) respectively.

The primary motivation for the development of Organization Theory is to gain understanding of the way human organizations operate and to use this knowledge as the basis for improving the design and performance of such organizations. Organization Theory is tightly coupled with Management Theory with respect to the control of the organization, with Psychology with respect to the behavior of the individuals, and with Economics with respect to the interactions between the organization and its environment (including the customers) and the way multiple organizations jointly affect the environment.

There are many approaches in Organization Theory to the modelling of human organizations. These approaches have focused on different aspects and/or components of organizations. The changing focus has lead to the emergence of different schools in the field. These could be identified as adopting a rational, natural, and open systems approaches [56, 96].

The Rational System approach is distinguished by two key assumptions that are related to the level of formalization of the goals and structure of the organization. It is assumed that the organization is primarily established to achieve goals. These goals are clearly defined and specific enough to provide a clear criteria among alternative means. It is also assumed that the organizational structure provides a clear definition of the roles, relationships between roles, and the rules that govern the behaviour of the participants (independently of the participants’ characteristics) (see for example March and Simon [76, 108] and Blau and Scott [7]).

The Natural System approach can be distinguished by the key assumption that the organization is primarily a collection of participants. Although the goals may be clearly defined they are not specific. It is assumed that the structures adopted are not formally specified but are developed according to the characteristics and interests of the participants. The structure can be seen as “naturally emerging” from the activities and abilities of the participants. The complexity of the goals and the importance of the informal structure is greatly emphasized (see for example Selznick [100, 101] and Parsons [80]).

The Open System approach can be distinguished by the key assumptions that the organization is primarily affected by the environment and that the participants do not necessarily share common goals. An organization is viewed as a changing coalition of participants. The organization does have a structure but is a system of interdependent activities. The level of dependency between the activities may vary depending on the activities, the organization, and the particular time (see for example Mintzberg [79] and Pfeffer and Salanick [81]).

Given that we are considering a computer system and given the requirements from the real-time embedded distributed system it seems that a combination of the Open and
Rational system approaches would be most relevant. This combined approach is a stream of Organization Theory referred to as the *Open Rational System* approach.

In the Open Rational System approach the goals and structure of the organization are very specific and highly formalized but the organization is embedded in the environment. Changes to the environment may affect the organization in a variety of ways, some of which are listed below:

- it may force the organization to adopt new goals or abandon old goals;
- it may affect the goals, preferences, and abilities of some or all of the participants; and
- it may affect the means available to the organization and the choice of the best means that should be employed to achieve the goals.

Command, Control, and Communication

The Command, Control, and Communication (C3) model [56, 64, 65, 73] has been primarily used to describe organizations that operate under mission critical circumstances. In particular they are used to describe military organizations. Due to the critical nature of such organizations they tend to adhere to a formal organizational structure and attempt to eliminate (through training and discipline) all informal aspects associated with natural human behavior.

C3 models are limited in the type of social relationships that they consider. As the name suggests, C3 models define organizational structures in terms of the command, control, and communication relationships that exist between the various units of the organization. Despite their relative simplicity there is still no single, clear, and agreed C3 model.

There are many definitions of command, control, and communication relationships [46, 95, 149]. Some of these definitions look both at the nature of the C3 relationships as well as the role of a commander, controller, and the underlying communication system. Nevertheless there is general agreement that in some form or another:

**Command** primarily involves the authority and responsibility to determine the objectives of the organization and the sub-units of the organization;

**Control** primarily involves the authority and responsibility for specification and modification of the detailed plans required for achieving the objectives and the monitoring of the execution of these plans; and

**Communications** primarily involves the sharing of information with respect to the state of the environment, the state of the organization, the state of the achievement of objectives, and the state of the execution of plans.

An organization can be described as a set of units or individuals with various relationships of command, control, and communication between them. These relationships are
with respect to different objectives, types of plans, and communication protocols. The specialization of the units of such an organization comes both from their knowledge and capabilities and their internal and external relationships.

In military organizations there is an attempt to reduce the impact of the human characteristics. The members of the organization are highly trained to follow “Standard Operating Procedures” and obey orders. In particular they include formal and rational behavior of the members and formal relationship between the units of the organization. Such characteristics are similar to the requirements we described above.

The research into and development of Command, Control, and Communication (C3) models have been heavily influenced by military organizations. It follows that C3 models are a good candidate to be used as a basis for developing a model of a distributed computer system.

1.4 An Overview of the Thesis

The work described in the following chapters incorporates and extends concepts from many approaches and from publications by many researchers. In particular, the approach to modelling real-time embedded distributed systems adopted here is primarily inspired by and based on a combination of the following work:

- The work on mathematical models of BDI agents by Rao and Georgeff [88, 90];
- mathematical models of distributed knowledge based systems by Halpern and Moses [54, 55];
- The work on philosophical aspects of joint activity by Tuomela [132, 135, 136];
- The work on models of human organizations by Mintzberg [79] and on Command, Control, and Communication systems by Harris and White [56] and Galley [46]; and
- The work on bounded rationality in human organizations by Simon [108, 110].
- The work on teams and multi-agent systems by Kinny et. al. [69] and Tambe [117, 119].

In the following chapters we introduce a hierarchy of models of distributed computer systems. Before we describe these models we provide information that is useful for the development of any new approach. This information includes an example domain which is used throughout this work to demonstrate the differences between the various models. The domain is air mission modelling and is described in detail in Chapter 2.

In addition to the example domain, we also introduce a detailed description of the functional requirements from a model of a distributed computer system. These requirements also refer to the support for the development of distributed computer systems, in particular the features of a specification language used to build such systems.
Given these requirements we also provide an overview of other approaches relevant to the specification and modelling of distributed computer systems. Such approaches include distributed artificial intelligence, distributed systems, organization and management theory, mathematics, and others. Both the requirements and relevant approaches are described in Chapter 3.

The hierarchy of models of distributed computer systems represent a series of abstractions and enhancements. Each of the models in the hierarchy generalizes the previous model. Each of the models allows the designer of the distributed system to explicitly specify more complex behaviors of the system and its components.

We start in Chapter 4 with a description of the single agent model developed by Rao and Georgeff [90]. We refer to this model as an agent-oriented model. In the agent-oriented model each component in the distributed system is modelled as an agent with ascribed knowledge about the environment. In particular we adopt a Belief-Goal-Intention (BGI) model of an agent. We refer to the beliefs, goals, and intentions of an agent as its mental attitudes. The mental attitudes of an agent are with respect to a particular state of the world.

A distributed system is composed of multiple agents. The explicit representation of the mental attitudes of the components of the distributed system allows the designer to validate and verify the specified behavior of these components [39, 55]. Although for each agent the environment includes other agents, the agent does not distinguish them and their actions from the environment.

Following the agent oriented-model we introduce in Chapter 5 the multi-agent model of a distributed system. In this model the actions and mental attitudes of different agents are distinguished by each of the agents in the system. That is, the actions and mental attitudes are now with respect to a particular agent and a particular state of the world.

This allows an agent to explicitly interact with, recognize and predict the mental attitudes and actions of, and react to the actions and mental attitudes of other agents in its environment. An agent-oriented system is regarded as a special case of a multi-agent system in which agents have no knowledge about other agents.

Following the multi-agent model we introduce in Chapter 6 the team-oriented model of a distributed system. In this model we change the basic model of a component of a distributed system from an agent to a team and define a sub-team relationship between teams. The mental attitudes of a team include mutual beliefs, joint goals, and joint intentions and are referred to as joint mental attitudes. The joint mental attitudes are with respect to a particular team and a particular state of the world.

A team which has no sub-team relationships with other teams is referred to as an agent. Given the environment is dynamic and may change over time we also allow the sub-team relationship to change over time. The set of sub-team relationships that a team has with other teams is referred to as the team structure for the team.

A team is a first class entity in the team-oriented approach. The change of focus from an agent in the system to a team allows the designer interested in the behaviour of the system as a whole to explicitly specify cooperation and coordination at a high level, without the need to consider all low level details. The way this specification is related to the detailed
behavior of the sub-teams can be described at a later stage. In brief, the team-oriented model supports a form of abstraction appropriate to distributed systems.

Unlike previous work [55, 69, 117, 135], the team-oriented model described here does not define the joint mental attitudes of the team in terms of the joint mental attitudes of the sub-teams. Rather, it is a primitive notion for the team. The designer can enforce the joint mental attitudes of the team to be in particular relationships to the joint mental attitudes of the sub-teams by imposing axiomatic or semantic constraints.

An advantage of this approach is that in a single framework one can model distributed systems with emergent behaviour, i.e., where the behaviour of the team is more than the sum of the behaviours of its individual team members, and also model compositional behaviour, i.e., where the behaviour of the team is merely a derived from the behaviours of the individual agents that constitute the team.

Abstracting the team’s joint mental attitudes from the sub-team’s joint mental attitudes also provides a framework for investigating different team structures and their impact on the team’s joint mental attitudes and team behavior. Furthermore, it provides a framework for investigating different types of joint mental attitudes for the team. A multi-agent system is considered to be a special case of a team-oriented system in which all the teams are agents. A number of axiom systems that explore the relationship between the joint mental attitudes of the team, the joint mental attitudes of the sub-teams, and the team structure are provided in Chapter 7.

Recall our assumption that the components of the system are required to interact. Although the team-oriented model focuses on a team it does not provide a clear model of the different types of relationships between the independent components of the distributed system. In particular there is no explicit model of the ability of one component to affect the joint mental attitudes of another component.

Following the team-oriented model we introduce in Chapter 8 the organization-oriented model of a distributed system. An organization-oriented model includes an explicit specification of the relationships between the various components. We refer to the components of the distributed system as organizations and to the relationships between them as social mental attitudes. The social aspects of the distributed system are inspired by organizational models from Organization and Management Theory [7, 76, 79, 96, 103] and in particular Command, Control, and Communication (C3) Theory [56, 64, 65, 73].

The organization-oriented model focuses on three types of relationships (or social mental attitudes) that exist between two organisations: (1) command; (2) control; and (3) communication. Unlike the joint mental attitudes, social mental attitudes are ternary relationships. That is, they are with respect to the two particular organizations and a particular state of the world.

The organization-oriented model described here does not define the social mental attitudes between two organizations in terms of their joint mental attitudes. Rather, this relationship is represented as a first class entity in our model. The designer can enforce the social mental attitudes between two organizations to be in a particular relationship to: (1) the joint mental attitudes of the two organizations; or (2) other social mental attitudes. This is achieved by imposing axiomatic or semantic constraints. A team-oriented model is
regarded as a special case of an organization-oriented model in which all the organizations have no social mental attitudes with other organizations.

An advantage of this approach is that in a single framework one can model different types of social phenomena, e.g., “blind obedience” in which a commanded organization always adopts as its own joint goal the joint goal adopted by the commanding organization; “dominant coalition” in which the joint goals adopted by the whole organization are determined by the joint goals adopted by the commanders of the organizations; etc. A number of such axiom systems are explored in Chapter 9.

A set of social mental attitudes between the sub-teams of an organization are referred to as the social structure for this organization. Given an organization, the combination of a team structure and a social structure is referred to as an organizational structure for the organization. An organizational structure is regarded as a means towards achieving the objectives of the organization. An organization can adopt multiple organizational structures based on its objectives and the various circumstances.

Although we have provided frameworks for specifying distributed computer systems we have not discussed the operative aspects of these specifications. That is, how these specifications translate into an operational model and how they affect the behavior of the system.

In Chapter 10 we provide the operational semantics for single agent, multi-agent, team-oriented, and organization-oriented systems. We particularly focus on the operational semantics of an organization-oriented system and provide a short analysis of the benefits and problems of this approach. Although we present a number of axiom systems we only describe the detailed operational semantics for one of them.

One important difference between using agents, teams, or organizations are the means available for achieving goals. In the agent-oriented approach the only means available to an agent are its own plans. In the team-oriented approach the means include the set of sub-teams and the possible allocation of tasks to these sub-teams. In the organization-oriented approach one also considers the way the set of sub-teams are organized in determining goals, making decision, and communicating information. These issues are also discussed in Chapter 10.

In a dynamic environment, one aspect of the operations of distributed computer system is the dynamic re-organization of the system into sub-systems, in particular, the formation and dismantling of teams and organizations. Before the formation of a team one has to first select from all possible teams the team that should be formed. This is not a simple task particularly under real-time constraints. In Chapter 11 we describe an approach for the selection of teams under the assumptions of limited time and computational resources.

The organization-oriented model of a distributed system provides the theoretical framework for modelling such systems. We refer to a distributed system that follows the organization-oriented approach as a organization-oriented system. In Chapter 12 we describe the design of a development environment for building organization-oriented systems. This design has been implemented as a prototype development environment. We refer to this environment as the TOP system.
1.4. AN OVERVIEW OF THE THESIS

The TOP system is an extension to the dMARS\textsuperscript{TM} system developed by the Australian Artificial Intelligence Institute. We provide an example of an organization-oriented system used for air mission modelling that was developed in cooperation with the Australian Department of Defence using the TOP system. This organization-oriented system simulates the behavior of multiple air defence forces engaged in an air combat mission.

The TOP system includes an organization-oriented programming language that allows the designer of the organization-oriented system to specify the desired structure and behavior of the distributed system. This language incorporates the different aspects of an organization-oriented system. The TOP system also includes a compiler and execution engine for the organization-oriented programming language. In Chapter 12 we also include an overview of an approach to software analysis and design that supports the development of organization-oriented systems. We conclude this work in Chapter 13 with an overview of the main contribution and a short discussion about possible extensions.

The work described above includes detailed mathematical models, system design, development approaches, and numerous examples. Some of this work has been published by us previously. Some of the ideas on coordination through the use of joint plans, as described in our discussion on operational semantics (Chapter 10), have been published in a paper by Kinny et. al. [69].

The ideas on teams and abstraction in a team-oriented approach, as described in our discussion on team-oriented systems (Chapter 6), have been published by us previously. Initial ideas on abstraction and encapsulation in multi-agent systems have been published in a paper by Cavedon and Tidhar [18]. The ideas on the use of teams and team knowledge for the specification of distributed systems have been published in a paper by Tidhar, Sonenberg, and Rao [131].

The ideas on social mental attitudes and social structures, as described in our discussion on organization-oriented systems (Chapter 8), have also been published by us previously. Initial ideas on social mental attitudes have been published in a paper by Cavedon, Rao, and Tidhar [15].

The work presented here on the approach to the selection of teams under time and resource constraints (Chapter 11) is an extended version of a paper by Tidhar, Rao, and Sonenberg [128]. The ideas and algorithms presented in Chapter 11 have been implemented and the results published in a technical report by Gabric et. al. [44]. A detailed description of the air mission modelling domain and an agent-oriented air mission modelling system have been described in a paper by Tidhar, Heinze, and Selvestrel [126].
Chapter 2

Example Domain

In this chapter we describe an example domain in which a real-time embedded distributed system is to be developed. The domain of application is Air Mission Modelling. In this domain the behavior of combat pilots in a situation of whole air missions is being modelled and analyzed.

The purpose of the modelling of air missions is to provide the defence forces with a better understanding and evaluation of the tactics used by combat pilots during air missions. We first describe the domain that is to be modelled and then provide an example of a particular air mission scenario. In the following chapters, we use the air mission modelling domain to highlight the benefits and limitations of the various models of real-time embedded distributed systems that are presented.

2.1 Air Missions

An air mission usually involves more than one aircraft from the same side. Aircraft are organized in particular formations and have specific roles assigned to them. For example, Howlett [58] shows how a mission to defend an air base might be accomplished by four sector CAPs (Combat Air Patrols) and two back-up CAPs. Each sector CAP consists of two aircraft with the designated roles of leader and wingman. The leader and wingman coordinate their actions between themselves and also coordinate their actions with the other sector CAPs.

Any mission involving multiple aircraft is accomplished by adopting organization tactics. The organization tactics are decomposed into sub-tactics based on the mission and the structure of the organization of pilots. Organization tactics give rise to notions such as mutual beliefs, joint goals, and joint intentions which are the beliefs, goals, and intentions held by multiple pilots or organizations.

The dynamic nature of air mission scenarios means that aircraft can dynamically reorganize themselves. For example, when two aircraft from different groups are shot down, the remaining aircraft of each group may combine together to form a single organization. Dynamic reorganization of aircraft may also force dynamic reassignment of roles.
The tactics of pilots involve carrying out a sequence of objectives or maneuvers. Each of these objectives can be further divided into smaller sub-objectives. For example, if the pilots decide to intercept a target they first need to find the bearing to the target; having determined the bearing to the target they need to change course towards it. Changing the course might involve a sequence of maneuvers. Thus, tactics are not just a set of rules, but a set of procedures that include complex (including sequential, parallel, and iterative) combinations of objectives and maneuvers.

At any instant the pilot might be able to employ a number of different tactics but may be executing only one of them. Also, while executing a particular tactic the pilot might have already decided (i.e., intended) to execute other tactics some time in the future.

The pilots need to continuously track the success or failure of their tactics as well as those of their adversaries; their very survival may depend on how well they are able to do this. The pilots thus clearly distinguish success and failure executions.

Information available from sensors, aircraft display, or inferred by the pilot, is called the perceived world, or set of beliefs, of the pilot. As well as beliefs the pilot also has a certain mission to accomplish, which is the primary objective of the pilot; from this objective arise other sub-objectives. To achieve these objectives, the pilot adopts a tactic. At any instant of time, the pilot will have multiple adopted tactics as to what is currently being performed or what will be performed sometime in the future.

In summary, any model of an organization of pilots must be powerful enough to account for:

- tactics with different combinations of objectives and maneuvers;
- having, executing, and adopting tactics;
- successful and failure executions;
- beliefs, objectives, and adopted tactics of pilots;
- organisations of pilots;
- roles of different organizations and pilots;
- mutual beliefs, joint goals and joint intentions of organizations of pilots;
- dynamic reorganisation of organizations and pilots; and
- dynamic reassignment of roles.

2.2 Air Mission Scenario

We will now describe a particular air mission scenario that the system has to model. We refer to it as the operational scenario. This scenario was originally described using terms and acronyms used by the Royal Australian Air Force. To increase readability we provide a description of the same scenario using terminology in more common use.
2.2. AIR MISSION SCENARIO

2.2.1 The Operational Scenario

The scenario includes two opposing sides. We will refer to them as Blue and Orange. Each side has different aircraft and ground based forces and equipment. Figure 2.1 illustrates such a scenario. Note that this figure is indicative only. A real scenario and setting can not be presented due to security reasons.

We start the scenario with a Blue Airborne Early Warning and Control (AEW&C) aircraft. This aircraft is given a task to patrol over water in a certain area at its normal operating altitude. That area is situated such that the AEW&C is required to detect and track enemy forces over both land and sea. Civilian aircraft and ships also cross this area on a regular basis.

The AEW&C is required to establish and maintain a surveillance picture on all enemy forces in a 360° sector around its own position. The surveillance picture is developed using on-board primary active radar, receivers for active Blue side identification signals, and passive detection using electronic measures.

Figure 2.1: An operational air mission scenario.
The AEW&C includes on board a Mission Commander (MC) and two Fighter Controllers (FC). The MC is tasked with controlling the activities of the AEW&C, coordinating the FCs. He will also command any fighter aircraft that may be assigned to him. This is done using the FCs. The FCs are tasked with monitoring the area and interacting with the assigned fighter aircraft.

The only other Blue force currently contributing to the surveillance picture in this area is an AP-3C maritime patrol aircraft. The AP-3C is given a task to identify and monitor a group of Orange ships. These ships may include civilian and military ships. The ships should be detected by the AEW&C.

A fourship of Orange strike aircraft with a pair of escort fighters enter the area that is monitored by the AEW&C. They are detected by the AEW&C’s primary radar. On first detection the Orange aircraft are automatically classified as unknown. The package does not generate Blue electronic identification signals and no electronic emissions are detected.

The detected forces are evaluated with reference to military intelligence information and are re-classified as unknown assumed hostile. The information about them is passed via satellite links to from the AEW&C to the Air Defence Ground Environment (ADGE) which controls the Blue anti-aircraft missile forces. At this time Orange forces commences jamming activity against the AEW&C from a transport aircraft, a C130, carrying appropriate jamming equipment. The C130 is operating beyond the detection range of the AEW&C. The AEW&C reacts by implementing appropriate electronic protection measures.

The activities of the of the Blue air forces are controlled by a Sector Air Defence Commander (SADC). The SADC assesses the situation and commands two pairs of Blue F/A-18 Hornet interceptors to establish a Combat Air Patrol (CAP) station. The two aircraft fly a course in the shape of a racetrack around the CAP station. This station is positioned to protect two high-value assets that might possibly be threatened by the Orange force. Two more pairs of F/A-18 Hornets are placed on the highest alert state on the ground.

The AEW&C mission commander (MC) is advised that the extended track of the Orange force will pass close enough to the AEW&C to necessitate a repositioning of the AEW&C station. This is done to ensure a minimum safe distance is maintained from any potential threat. The MC initiates a repositioning and advises the Air Defence Ground Environment (ADGE).

At this stage, a pair of aircraft detaches from the Orange fourship, flying in the general direction of the AEW&C. The MC assesses the threat that the detached pair imposes on a Blue Reporting Point deployed on an island. A Reporting Point typically consists of a Ground Based Radar (GBR) with communications facilities.

The AEW&C itself could also be threatened. The MC commands another repositioning towards the nearest CAP station and advises the Sector Air Defence Commander (SADC) of the new situation. The SADC commands the pair of F/A-18 Hornets on alert to establish a CAP station that provides protection to the Reporting Point and the AEW&C.

The SADC assigns these Hornets to the operational control of the AEW&C MC. The control and reporting communications and tactical data exchange between the MC and the Hornets is achieved via secure, jam-resistant tactical data links and voice communications.
The remainder of the Orange package crosses the Australian coastline, turns to fly along the line of the coast and descends to low altitude. The Orange escort fighters activate their radar in search mode. These emissions are detected and analysed by the electronic detection system on-board the Blue AEW&C. The Orange force is re-evaluated and assessed as hostile, posing a potential threat to a coastal Reporting Point.

The additional electronic detection data indicates that there are two Orange escorts. The tracks are passed to the ADGE. The SADC commands the pair of Hornets on the closest CAP station to intercept the package. The AEW&C MC is also assigned operational control of this pair.

When the MC is reasonably certain that the Blue coastal Reporting Point is the target of the Orange strikers, the MC assigns a Fighter Controller (FC) to control the intercept. The SADC scrambles another two pair of F/A-18 Hornets to occupy CAP stations protecting high-value assets on the mainland.

Suddenly the two detached Orange aircraft activate their radars in search mode, revealing themselves. Analysis of electronic emissions reveals to the MC that these are fighters. They are immediately classified as hostile and are assessed as a direct threat to the AEW&C. It is still not possible, however, to determine if the intended target of this pair is the island Reporting Point or the AEW&C itself.

The SADC assigns the pair of F/A-18 Hornets on the closest CAP station to the MC for self-protection of the AEW&C aircraft. The MC assigns a second FC to control this pair and to initiate an intercept of the Orange fighters. The MC also initiates a repositioning at maximum speed toward the protection of the Hornets.

The AEW&C is now required to manage two air defence tasks: a high-altitude intercept to its rear and another intercept on its at low altitude near the coastline. Both Orange pairs contain aircraft with jamming equipment that are capable of disrupting the Hornet radar. All four Hornets are thus almost completely dependent on the AEW&C for their tactical information and air picture.

The FCs guide the Hornets to achieve an intercept, monitor the progress of the engagement and advise the Hornet pilots. They monitor the Hornets’ fuel and weapons state and advise the ADGE. They provide information on threat disposition and tactics if requested or if their assessment of the pilots’ situational awareness indicates that it is required.

The way the scenario continues to unfold and its outcome may depend on the exact starting conditions, the behavior of the operators, and the capabilities of the equipment. Multiple simulations will provide the basis for a statistical analysis of how will this engagement end.

### 2.2.2 Analysis

The air mission scenario example includes a number of organizations that adopt various organizational structures under different circumstances. In particular the relationships between the fighter controllers and the fighter pilots change over time.

We will first describe the organizations and agents that are part of this example and the sub-team relationships between them. We will then proceed to describe the various
2.2. AIR MISSION SCENARIO

Organizational structures that these organizations can adopt. We will conclude with a description of the organizational plans that they employ in achieving their objectives.

Organizations and Agents

In the above example there are two primary organizations: Blue force and Orange force. We will focus here on the Blue force and the various sub-teams which are involved in its C3 structure.

At the initial stage the Blue force has as sub-teams a Sector Air Defence, AEW&C aircraft, AP-3C aircraft, and ADGE. The Sector Air Defence has as sub-teams a SADC, and 2 pairs of hornets. We refer to these pairs as an Attack and a CAP respectively. The SADC is an agent and each hornet in the pair is also an agent. The AEW&C has as sub-teams an MC and two FCs. In this scenario we view the AP-3C and ADGE as agents. It follows that we need to specify the agents:

- AEW&C organization;
- Attack organization;
- CAP organization;
- a single SADC agent;
- four hornet agents;
- a single MC agent;
- two FC agents;
- a single AP-3C agent; and
- a single ADGE agent.

It is important to note that as described above the organizations and relationships between them change over time. In particular the two FCs on-board the AEW&C each join the Attack and CAP to form two new organizations. The two organizations are joined by the MC to form the new AEW&C organization.

Organizational Structures

We focus on the relationships between the organizations and agents in the Blue force. In particular we will focus on the the AEW&C and its the way they relationships change over time.

As mentioned above at the initial stage the AEW&C includes three sub-teams: an MC and 2 FCs. The MC is the commander and controller of the two FCs. That is, it can provide them with goals and determine how they will achieve these goals. On the inclusion
of the 4 hornets each FC has a special relationship with a pair of hornets. The hornets also have special relationships between them.

Let us consider the possible relationships between one of the FCs and one of the pairs, the pair that is tasked with the attack of the enemy. We refer to the group as Attack. The relationships depend on the tasks (or goals) the AEW&C is attempting to achieve and the particular circumstances under which it is attempting to achieve these tasks. In particular the relationships will depend on the level of autonomy that could be exercised by the pair of hornets. This autonomy will depend on the information available to them (e.g., radar, visuals, etc.), the reliability of the radio communication, and other parameters. If the Attack pair can operate relatively autonomously then the relationships between the FC and the hornets will include only command and communication. We say that they have a “tactical control” relationship.

If the Attack can not operate relatively autonomously then the relationships adopted by Attack will also include control relationship between the FC and the pair of hornets. This control will only be with respect to the choice of tactics adopted by the attack. Such type of relationships are referred to as “close control” relationships.

Note that the way these organizations are implemented will depend on the particular approach adopted. Similarly the way the relationships between the agents and organizations are implemented will also depend on the approach adopted. In the following chapters we present different approaches to representing these organizations and organizational structures.

Key Challenges and Relevant Characteristics

The air mission modelling domain and the particular scenario include the following key challenges:

**Dynamic:** The environment in which the various forces operate is dynamic and changes while the mission is being performed. Furthermore the elements of the simulation are required to adapt their behavior and relationships in response to these changes. Such changes include changing organizational structures as well as role assignment within the same structure.

**Real-Time:** The elements of the air mission must respond to the changing environment under real-time constraints. This is further emphasized when the system is used interactively by air operations analysts and domain experts.

**Distributed:** The complexity of the tasks performed by each of the components in the scenario and the complexity of the air mission requires the use of multiple components. These elements of the air mission are required to cooperate and coordinate to achieve joint. They do so by using team tactics that are known to all members of the team.

The behavior of the elements of the air mission and the relationships between them are highly formalized and well documented. They include:
Multiple Goals: At any given time the elements of the air mission may possess more than one goal. As an example, the AEW&C is required to monitor an area and support the Hornets in their intercepts.

Common Goals: The elements of the air mission share common goals. Such goals include defending an area, attacking an area, intercepting the AEW&C, etc.

Social Rationality: As elements have both individual and joint tasks there is some relationship between these; for example, eliminating a ground target (i.e., the mission goal) vs. shooting down another enemy aircraft. The relationships between the individual, joint mental attitudes, and social structure will depend on the characteristics of the element. Individualistic pilots may apply their own judgement and decide to ignore the instructions from the SADC (and the joint goal) in favour of their individual goals.

Heterogeneous Ability: The elements of the air mission have varying abilities some of which may be overlapping. Such abilities include: (1) the ability to perform a task (e.g., eliminate a target); (2) the ability to execute a tactic (e.g., perform a pincer intercept on a target); and (3) the ability to play a role within a team (e.g., a Mission Commander).

Organizational Ability: The organizational knowledge and the ability to manipulate it exists in all elements. Furthermore the air mission domain requires that the elements use these abilities in determining their behavior and in response to the changing circumstances.

Bounded Ability: The elements of the air mission have bounded abilities. A computer system that models such elements is also bounded and will be limited in its ability to model all elements.

Communication: The elements of the air mission have radio, data link, and satellite communication facilities available to them.

To conclude we believe that the air mission modelling domain serves as a good representative of the types of problems we are addressing in this work. In the following chapters we will use this domain to highlight different aspects of the proposed approach.
Chapter 3

Required Features and Relevant Approaches

The work described in following chapters is focused on developing a model of organizations of artificial agents, we refer to such organizations as artificial organizations. The work attempts to use and adapt concepts and ideas from OMT and show how such ideas can be used in designing artificial organizations. By no means does this work attempt to present new ideas for OMT. Nevertheless, we hope that in some way it will also shed some light on these concepts from a different perspective, the perspective of multi-agent systems.

In this chapter we provide a detailed description of the requirements of a model of distributed real-time embedded computer system. Following this description we provide an overview of related models from Organization and Management Theory, philosophy, theoretical computer science and distributed systems, and artificial intelligence. We also provide an analysis of the limitations of these approaches to the problem addressed here.

3.1 Requirements

As described in the introduction, our main objective is to investigate and develop real-time embedded distributed systems. In particular we are interested in distributed systems that coordinate the activities of their independent components. We are interested in an approach that will allow researchers and developers to:

1. precisely specify a real-time embedded distributed system;

2. investigate different types of system behavior;

3. most importantly, use a specification language that is expressive enough to cover all aspects of the real-time embedded distributed system as described in Chapter 1.

As described by Emerson [36], one could use a formal language as a specification of a computer system. This language will provide a mathematical model and proof theory to
3.1. REQUITEMENTS

reason and make statements about a real-time embedded distributed system. We refer to
this language as a formal syntactic model of a real-time embedded distributed system.

The meaning of these statements must be given by a semantic model, that is, a meaning.
The semantic model would provide the interpretation of the syntactic statements. To
ensure that this interpretation is precise one would require a formal semantic model of a
real-time embedded distributed system.

Given the semantic and syntactic models one is faced with the problem of precisely
identifying the relationship between the two models. In particular, one should be able to
specify a condition in the semantic model and guarantee that the appropriate statement
in the syntactic model will hold and vice versa. That is, we require that the semantic and
syntactic models of a real-time embedded distributed system are sound and complete with
respect to each other.

In this section we describe the functional requirements of a formal language for spec-
ifying real-time embedded distributed systems. In particular we focus on the specifi-
cation of real-time embedded distributed coordination and on a language for specifying
Organization-Oriented Systems. We refer to this language as an organization-oriented
specification language. In such a language there are basically two issues that need to be
addressed. The first is the different types of knowledge and their representation. The
second issue is the process of reasoning and manipulation of this knowledge.

Furthermore, one has to identify the potential states of the system. For each such state
one has to specify the expected behavior of the system. We identify the following types of
system states:

• an organization is in a state of the world in which another organization has to be
formed; for example, if it jointly desires to achieve another state (i.e., a joint goal
state) but can not, and a new organization can possibly achieve that joint goal state.
This situation requires a behavior resulting in a new organization being formed. We
refer to this behavior as Organization Formation. A similar state is a state in which a
group of sub-teams should stop behaving as an organization. This situation requires
a behavior resulting in an existing organization being dismantled. We refer to this
behavior as Organization Dismantling.

• an organization is in a state of the world which requires it to act. This can either
be because of new data that has arrived or because the organization has a new goal
to achieve. This situation requires a behavior in which the organization responds to
the new information. We refer to this behavior as Organizational Planning.

• given the above states, an organization is in a state in which it has multiple options to
choose from. For example, multiple joint goals to achieve, multiple means to achieve
a goal, multiple sub-teams to achieve a sub-goal, etc. We refer to the behavior of
choosing from multiple options as Organization Deliberation.

The way the behavior of organization is defined is directly affected by the way the
world is modelled and the way organizations are represented. The language should allow
the specification of structured organizations. That is, the developer should be able to specify the relationships between the various sub-teams of an organization.

Since this is a specification language, for each type of specification one should also define the way these specifications are acted upon, i.e., its operational semantics.

3.1. Interactions Between a System and its Environment

The organization-oriented specification language should allow for the specification of embedded systems. By definition an embedded system is a system that interleaves perceptions, actions, and re-actions. It follows that the specification language should allow for the specification of the possible states of the environment and a description of the possible states of the organizations.

Given that an organization is required to react to the changing environment it must have as part of its state a model of this environment. We refer to this model as “the organization’s model of the world”. For each organization the specification language should allow the developer to specify which parts of the world are perceived by it and which parts are included in its model of the world.

An organization can affect the world by performing an action. The organization-oriented specification language should allow the developer to specify which actions are available to which organization. Furthermore the developer should specify how these actions affect the environment. In addition, the specification language should allow the developer to specify actions that change the internal state of the organization.

Given that the environment and the system are changing over time, the developer should be able to specify which states occur at what points in time. That is, how the environment and the system change over time. Note that because we have discrete states we only require a specification of discrete time points.

3.1.2 Representing Organizations

The language should define the notion of an organizational structure adopted. Such a definition should allow the developer to specify the organizations that are part of an organization (i.e., Team Structure), inter-organization relationships (i.e., Social Structure), the function assigned to each sub-team (i.e., Responsibilities), and the way the organization can be referred to or represented by other organization (i.e., Identification). Furthermore the notions of the beliefs of an organization (i.e., Mutual Beliefs), the goal states desired by an organization (i.e., Joint Goals), and the organizations intention towards achieving these goals (i.e., Joint Intentions) should all be defined.

Components that support the reasoning processes that are executed by the organization are also part of the organization definition. These include the skills of the organization used for organization formation.

Organizational Structure: One of the most dominant features of organizations in the real world is the fact that they have a structure which guides the way the organization
operates. This structure defines the way data and decisions flow within the organization. Furthermore, the structure of the organization defines the sub-teams. The language should allow the developer to define organizations with varying such structures. The language should also describe how the structure of the organization affects the behavior of the organization.

**Responsibilities**: Part of the definition of the organization are the responsibilities that are allocated to different members of the organization. Such responsibilities can be grouped into two categories that correspond to responsibilities regarding: (a) the function of the set of sub-teams or agents as an organization; and (b) the function of the organization in the domain. We refer to (a) as Social Responsibility and to (b) as Domain Responsibility. One can attach a tag to these responsibilities, we refer to this tag as a role and will use the terms Social and Domain Role as tags for Social and Domain Responsibilities respectively. The language should allow the developer to specify how are the responsibilities assigned to the different sub-teams.

**Identification**: Humans refer to organizations in many ways: (1) a label or name which identifies the organization, e.g., AAAI; (2) a person that represents the organization, e.g., the director; (3) a goal which the organization is currently pursuing, e.g., the company that is painting the house; or (4) the plan which the organization is using, e.g., the guys using the ladder. The language should provide a mechanism for naming an organization and referring to an organization by its name.

**Mutual Beliefs**: An organization has beliefs about the current state of the world. We refer to these beliefs as mutual beliefs. The developer should be able to specify the types of mutual beliefs that an organization can hold. The relationships between the mutual beliefs of the organization and the mutual beliefs of the sub-teams should be determined by the developer. The language should describe how mutual beliefs are modelled.

**Joint Goals**: An organization has states of the world that it desires to be in. We refer to these states as joint goals. The developer should be able to specify the types of joint goals that an organization can desire. The relationships between the joint goals of the organization and the joint goals of the sub-teams should be specified by the developer. The language should describe how joint goals are modelled.

**Joint Intentions**: The overall behavior of the organization is a sequence of executions of actions that take it from one state of the world to the other. This sequence can be described as a combination interleaved executions of parts of the sequence (i.e., sub-sequences). Each sub-sequence is a combination of states that start from a current state and end at a desired state. We refer to such a combination as a joint intention. An organization adopts joint intentions in an attempt to achieve its joint goals. Different intentions could be used to achieve the same joint goal starting from different states. The developer should be able to specify the types of joint intentions.
that an organization can adopt. The relationships between the joint intentions of the organization and the joint intentions of the sub-teams should be specified by the developer. The language should describe how joint intentions are modelled.

**Skills:** The skills of an organization are the set of behaviors it can potentially exhibit. An organization can potentially exhibit a behavior if it can form a joint intention corresponding to that behavior. The developer should be able to specify the types of joint intentions that an organization can adopt. The relationships between the skills of the organization and the skills of the sub-teams should be specified by the developer. The language should describe how the skills of the organization are modelled.

**Social Knowledge** An organization has knowledge of the organization in its environment and its social relationships with these organizations. These relationships are referred to as *social mental attitudes*. In addition the organization may be part of another organization. As such it may have responsibilities in the context of the other organization. The developer should be able to specify what type of social knowledge is available to the organization. Furthermore the developer should be able to specify how this knowledge affects the mutual beliefs, joint goals, and joint intentions of the organization.

### 3.1.3 Organization Formation and Dismantling

There are many reasons why new organizations should be formed. These include a redistribution of the current problems that are addressed, new problems that are introduced, and re-organization of the sub-teams, just to name some of them. There may also be similar reasons why existing organizations should be dismantled. The reasoning required for choosing the organization to be formed or dismantled is discussed below as part of the discussion on Organization Decision Making. Here we discuss the problem of forming and dismantling an organization once the organization to be formed or dismantled has been chosen.

One should be able to specify the exact protocol to be used by an organization and the potential sub-teams of the new organization when forming the organization. This includes the synchronization of the mental attitudes of the sub-teams and determination of the mental state and abilities of the new organization. Similarly one should be able to specify how does an organization dismantle.

**Knowledge Representation**

**Sub-Teams:** In-order to form (or dismantle) an organization the first type of knowledge is the set of organizations that will be associated with the organization in the case of organization formation, and the organization itself in the case of dismantling. The set of organizations that are associated with the organization are referred to as the *sub-teams* of the organization. The sub-team relationship between two organization
is expressed in terms of the mental state of the two organizations, in particular their mutual beliefs.

**Means and Ends:** Organizations are typically formed to achieve a particular goal. One should be able to specify towards which end is the organization formed. It may also be the case that the organization is formed towards particular ends using particular means, again one should be able to specify which means should be used. These specifications may affect the way the organization is formed and the success of the organization formation process.

**Mental Attitudes:** The social and joint mental attitudes held by the organization are based on social and joint mental attitudes of the sub-teams. One outcome of the organization formation (or dismantling) process are the adoption (or removal) of the respective social and joint mental attitudes by the relevant sub-teams.

**Skills:** Like mental attitudes, the skills of the organization may depend on the skills of the sub-teams. These are determined when the organization is formed and are an outcome of the organization formation process.

**Reasoning**

**Mind-Set Synchronization:** One aspect of the formation (or dismantling) of an organization is the change to the social and joint mental attitudes of the sub-teams. This includes the addition (or removal) of social knowledge, mutual beliefs, joint goals, and joint intentions. We refer to this process as *mind-set synchronization*. The developer should be able to specify which mental attitudes should be synchronized and how.

**Skill Creation:** Part of the definition of an organization is the skills of that organization. These skills are typically composed of the skills of the sub-teams. The developer should be able to specify how the skills of the organization are created. The developer should also be able to specify how the skills of the sub-teams should be combined to form the skills of the organization.

**3.1.4 Organization Decision Making**

There are many situations in which an organization is faced with multiple options. These situations include the case of multiple goal states to be achieved, multiple organizations that can achieve a goal state, and multiple intentions that achieve a particular goal state. Each such case requires the organization to decide on the preferred option. As mentioned before, this behavior is referred to as *Organization Decision Making*.

There are different ways in which the organization can make a decision, e.g., decision trees, negotiation, voting, etc. The language should allow the developer to specify the exact way in which decisions should be made by the organization.
3.1. REQUIREMENTS

Making a decision regarding each such option requires specific type of knowledge and reasoning, for example, the choice of organization for a particular goal. The knowledge required for such type of reasoning involves the knowledge of current existing organizations, their skills, and their capabilities. While skills are static in nature and describe what an organization can potentially do, the term capabilities refers to when an organization can do so. For example, an organization of 4 fighter aircraft can potentially intercept a pair of fully armed enemy strike aircraft. Taking into account the current fuel level of the 4 fighters will mean that the fourship can not actually intercept the enemy aircraft.

The actual way that goals, plans, and organizations are represented in order to facilitate organization decision making may be different from the way they are represented for other purposes such as organization formation and plan execution. The language should specify how these notions are defined for organization decision making.

Knowledge Representation

Joint Goals: If an organization has more than one goal simultaneously it has to decide on the order in which it should attempt to achieve these goals or, even simpler than that, the next goal that it should attempt to achieve. In order to be able to choose the next goal the list of current goals should be accessible.

Organizational Plans: After a goal has been chosen, an intention is to be formed. The formation of intentions should adopt the pre-planned approach. We refer to each such plan as an organizational plans. An organization may have more than one plan that could be used to form an intention for achieving that goal. The organization has to select the intention to be formed (i.e., the plan to be used to form the intention).

Current Organizations: When the organization is required to form an organization it either needs to know what are all the current organization that are potential sub-teams for the new organization or it needs to have access to a facility that will provide this information, e.g., communications, server, etc.

Organizational Structures: An organization may adopt different organizational structures in attempting to achieve its joint goals. There may be multiple organizational structures that the organization can adopt. The organization has to select the organizational structure to be adopted.

Skills: For each of the above organizations the skills that it possesses should be accessible to the organization making the decision. Since the skills of an organization to be formed (or dismantled) depend on the skills of the sub-teams, this information is required for making such a decision.

Capabilities: For each of the above organizations the capabilities that it possesses should be accessible to the organization making the decision. Since the capabilities of an organization to be formed (or dismantled) depend on the capabilities of the sub-teams, this information is required for making such a decision.
3.1. REQUIREMENTS

CHAPTER 3. REQUIRED FEATURES AND RELEVANT APPROACHES

Reasoning

Choosing Joint Goals: The developer should be able to specify how the organization chooses a particular goal when there are multiple goals to choose from. The language should allow the developer to specify mechanisms for organization decision making such as voting and negotiation.

Choosing Organizational Plans: Similar to choosing joint goals, the developer should be able to specify how the organization chooses a plan when there are multiple plans that can be used to achieve a particular goal.

Choosing Organizations: When the organization is in a state in which it has to form (or dismantle) an organization (see Section 3.1.3), it has to make a decision as to which organization should be formed (or dismantled). This can be done using different methods and they typically will depend on the skills and capabilities of the potential sub-teams. The developer should be able to specify how the organization makes this choice.

Choosing Organizational Structure: Similar to choosing joint goals, the developer should be able to specify how the organization chooses an organizational structure when there are multiple such structures that can be used to achieve a particular joint goal.

Role Assignment: Given a particular organizational structure the assignment of responsibilities to the sub-teams is required. We refer to this process as role-assignment. The organization has to decide on a particular role assignment. The developer should be able to specify how the organization makes such a decision.

Responding to Failure: An attempt to act on any of the decisions made by the organization may fail. The developer should specify how the organization responds to this failure.

3.1.5 Payoffs, Costs, and Probabilities

The above requirements include the specifications of various types of deliberation and decision making processes. These processes were primarily qualitative. They did not include any quantitative measures that are normally associated with decision theoretic type processes. The types of quantitative measures would include payoffs, costs, and probabilities.

Rao and Georgeff [86] have shown how a qualitative approach for deliberation about the selection of goals and intentions could be generalized to use quantitative measures. We believe that similar techniques could be used to introduce quantitative measures to the decision making processes but we do not deal with these issues here.
3.2 Related Approaches

In this section we describe approaches from Organization and Management Theory, Computer Science, Distributed Systems, Artificial Intelligence, and Software Engineering that are relevant to the specification and development of artificial organizations. We will briefly describe why these approaches are relevant. In the following chapters we will describe, where relevant, which concepts from the following approaches were utilized.

3.2.1 Models of Organizational Structures

When observing the behavior of a group of humans that have been assembled together to achieve some particular objective one can immediately recognize some forms of structures that the group has adopted. Such structures may include informal roles such as a leader or planner or formal roles such as project manager or company accountant. The structures may also include the way the different roles interact and the dependencies between them.

The field of Organization and Management Theory (OMT) has been mainly focusing on conceptualizing and formalizing the notions of human organizations and organizational behavior [76, 103] and on creating theories that will assist in designing such organizations [116]. Since OMT has considered human organizations, a substantial effort has been directed towards understanding the effects that the human nature and behavior have on the design of human organizations and the understanding of the relationship between the formal and informal organizations [7, 121].

It is widely accepted in the OMT community that the major reason for the formation of an organization is the bounded ability of people to handle complex tasks and to process vast amounts of information. These limitation have been referred to as bounded rationality [76]. As organizations involve multiple people one has to decompose the tasks of the organization into sub-tasks, assign them to the different members, and coordinate their execution. To structure the continuous task decomposition, task allocation, and coordination the organization includes roles and tasks that perform these activities and facilitate the functioning of the organization as a whole. These roles and tasks can be mainly categorized under two type of processes, communication processes and control processes [12, 140].

The work described here is focused on developing a model of organizations of artificial agents, we refer to such organizations as artificial organizations. The work attempts to use and adapt concepts and ideas from OMT and show how such ideas can be used in designing artificial organizations.

Furthermore, we extend the notion of bounded rationality of artificial agents to bounded rationality of artificial organizations. The notion of bounded rationality with respect to the decision process is highly related to the notion of uncertainty. This is uncertainty about the the current state of the world (i.e., limited knowledge) as well as uncertainty about the outcome of actions. Because of the bounded ability to reason there is a limit amount of information or to the number of options that each individual or group are able to consider.

Some aspects and complexities of organizations are a direct outcome of human nature, which can be ignored in artificial systems. Such aspects include, amongst others, obedience
of subordinates, motivation, and emotions. On the other hand, other aspects have to be more rigorously defined. Such aspects include, amongst others, rationality, complexity, and utility of organizations. In order to clearly identify the OMT concepts that are used and those that are ignored we present a short overview of the basic concepts commonly used in OMT and explain why we believe these concepts are either relevant or not to building artificial organizations.

The idea of using concepts from OMT for building DAI systems has been first put forward by Mark Fox [41, 42] and his original work in this area is probably one of the most referenced works on organization structures within the DAI literature. As stated by Mark Fox:

“The approaches described by organization theory are interesting and useful but not rigorous. Better methods of measuring complexity and uncertainty must be found. Whether these measures will be derived from organization theory, system science, or computer science remains to be seen.”

M. S. Fox, *An Organizational View of Distributed Systems* [42]

We hope that this thesis is a step in the right direction and hope that it will provide a more rigorous analysis of organizations in the context of artificial systems.

OMT models that focus on the formal aspects of organizations, e.g., structure, formal roles, formal responsibilities, rationality, etc. are referred to as formal organizational models. As mentioned above in dealing with artificial organizations we are primarily interested in such models. Let us now provide an overview of different formal organizational models.

### 3.2.2 Formal Organizational Models

In this section we describe different approaches that have been previously adopted in describing formal organizations. Most of these approaches have stemmed from OMT. In general a formal organization is one that only provides the formal roles, responsibilities, structure, or communication channels between the different elements in the organization. It does not take into account the aspects of the organization that are related to the personal relationships that exist between individuals or the effect of human personality on the organization. Such aspects may include charisma, fear, social groups, etc. Let us first provide definitions of general organizations and formal organizations.

An organization provides a framework to the way in which a group of agents (human or artificial) interact. The different interactions between the members of the group can be categorized as different social relations. In the most general terms an organization as defined by Blau and Scott [7] consists of: (1) a structure of social relations in a group of agents; and (2) the shared mental attitudes and orientations that unite the members of the group. This definition is expanded by Scott [96] to include the following components:

The **Participants** are the individuals that make contributions to the organization in return for a variety of inducements.

---

1This definition is based on an earlier definition by Leavitt [71].
The Goals of the organization are the desired ends and the purpose of the organization.

A Social Structure refers to the regular aspects of the relations and interactions among the participants.

The Technology includes the means that are available to the participants for achieving the goals.

An organization is also embedded within a particular environment. This environment may include other organizations and indeed the organization can actually be part of a larger organization, e.g., a company may be part of a large consortium. The nature of the environment will impact the goals, social structure, and technology used by the organization. For example, an organization which is embedded in a highly dynamic environment will have to rapidly react to the changes in the environment. The control processes used to make decisions will have to be very efficient and the time taken to make a decision relatively short.

A formal organization is an organization in which all four components and in particular the social structure have been formally specified. That is there has been a clear and precise specification of all the participants, all the goals of the organization, all aspects of the social structure, and all the means employed by the participants in achieving the goals. Humans are very complex and our understanding of them is limited. It seems that in human organizations there are some aspects that are very difficult to formally specify. Such aspects include the personal relationships between the participants, social norms that guide the participants, and the different goals that some sub-groups may have, etc. Such aspects are referred to as the informal organization.²

Given our limited understanding and control of human behavior it seems that in human organizations it is impossible to completely specify and control the emergence of informal organizations. For the same reason it is also very difficult to identify all the mutual influences that exist between the formal and informal organizations. The work described here focuses on groups of artificial agents and we assume that there are no informal organizations and that all aspects of the organizations are (and possibly should be) formally specified. Hence we only consider formal organizations.

History and Classification

Given the above definition of an organization there have been a variety of approaches that have focused on different aspects and/or components of organizations. The models developed have been described at varying levels of formality. The changing focus has lead to the emergence of different schools within the OMT community. These could be identified as adopting a rational, natural, and open systems perspectives [96].

²It is interesting to note that as the understanding of human behavior grows the definition of what are the formal and informal aspects changes.
Rational Systems approach is distinguished by two key assumptions that are related to the level in which the goals and structure of the organization are formally defined. It is assumed that the organization is primarily established to achieve goals. These goals are clearly and formally defined. Furthermore, it is assumed that these goals are specific enough to provide a clear criteria among alternative means of achieving these goals, i.e., the technology. It is also assumed that the social structure provides a clear and formal definition of the roles, relationships between roles, and the rules that govern the behaviour of the participants (independently of the participants’ characteristics and goals).

Researchers in OMT that can be categorized as belonging to the rational systems school include Taylor [120], Weber [141], March and Simon [76, 108], Blau and Scott [7], and Etzioni [38]. Although their approaches are not identical, it is identified by Scott [96] in his analysis that they share the view that “organizations are collectives oriented to the pursuit of specific goals and exhibiting relatively highly formalized social structures”.

Natural Systems approach can be distinguished by the key assumption that the organization is primarily a collection of participants. Although the goals maybe clearly and formally defined they are very high level. That is, the goals provide general objectives which are shared by the participants but are beyond the specific means (i.e., the technology) available to the participants. It is further assumed that the structures adopted are not formally specified but are developed according to the objectives, characteristics, and interests of the participants and the constraints introduced by the environment. The structure can be seen as “naturally emerging” from the activities and abilities of the participants and in response to the changing environment. The abstract nature of the goals and the importance of the informal structure is greatly emphasized.

Researchers in OMT that can be categorized as belonging to the natural systems school include Mayo [78], Barnard [3], Selznick [100, 101], and Parsons [80]. Again, although their approaches are not identical, it is identified by Scott [96] in his analysis that in general they share the view that “organizations are collectivities whose participants share a common interest in the survival of the system and who engage in collective activities, informally structured, to secure this end”.

Open Systems approach can be distinguished by the key assumptions that the organization is primarily affected by the environment and that the participants do not necessarily share common goals. An organization is viewed as a changing coalition of participants. The organization does have a structure but this structure is more a collection of interdependent activities than formal roles and relationships between these roles. The level of dependency between the activities may vary depending on the activities, the organization, and the particular time. The open systems approach has been mainly influenced by the work on general systems founded by Bertalanffy [140].

Researchers in OMT that can be categorized as belonging to the open systems school include Galbraith [45], Weick [142], Mintzberg [79], Pfeffer and Salanick [81], and March...
and Simon [76]. Again, although their approaches are not identical, it is identified by Scott [96] in his analysis that in general they share the view that “organizations are systems of interdependent activities linking shifting coalitions of participants; the systems are embedded in - dependent on continuing exchanges with and constituted by - the environments in which they operate”.

Although the above classification suggests three distinct approaches the rational systems and natural systems share an underlying common assumption. Both approaches assume that the organization and any external events that affect it are completely known. Systems making such an assumption are categorized as closed systems. If one adopts a high level classification of the trends in OMT one can classify these trends into a general (and crude) change in emphasis and approach over the past century. Until the middle of the 20th century a closed system model has been dominant. Since the 1960s the open systems model has become dominant. Under these two models there has been a change of focus from rational to natural systems. The current dominant model seems to be an open natural system model.

The change in the dominant models has not only occurred because of the change in the perception of the researchers developing OMT. The change in the environment in which organizations are required to operate has had a major impact on the development of OMT. Amongst other reasons, the availability of information and ease in which participants can and do change organizations have promoted the development of the open system model.

Although the open rational system (ORS) model is not the currently dominant model, given the assumptions in Chapter 1 it is the best suited model (with some variations) for developing a theory of artificial organizations. Hence in the following section we will provide further details on the development of open rational systems (ORS) in OMT.

Open Rational Systems

Recall that the basic assumptions of the rational systems approach is that the goals and structure of the organization are very specific and highly formalized. Furthermore, the basic assumptions of the open systems approach is that the organization is primarily affected by the environment and that the participants do not necessarily share common goals.

In the combined model, the open rational systems (ORS) model, the goals and structure of the organization are very specific and highly formalized but the organization is embedded in the environment. Changes to the environment may affect the organization in a variety of ways including:

- forcing the organization to adopt new goals or abandon old goals;
- forcing some or all of the participants to change their goals, preferences, and abilities; and

\(^3\) Although March and Simon have been identified as belonging to the rational systems school they have also contributed to the open systems school and some parts of their work can be viewed as a combination of the two approaches, i.e., as open rational systems.
affecting the technology available to the organization and the choice of the best
technology that should be employed to achieve the goals.

Within the ORS model there have been a number of streams of research each adopting
a different view of organizations. Although they agree on the basic underlying assump-
tions they suggest different explanations to the reasons for the emergence of organizations,
the function of an organization, and the important aspects and activities within an or-
ganization. Such streams include the focus on bounded rationality by Simon [108, 110],
transaction costs analysis by Williamson [145], and contingency theory by Thompson [121].

Note that all ORS models share the assumption which is also made here that the partic-
ipants of an organization are bounded in their cognitive abilities. Nevertheless different
ORS models put different emphasize on this feature.

The structure adopted by the organization is affected by the goals, participants, tech-
nology, and the environment and the interdependencies that exist between them. These
interdependencies will typically lead to the need for coordination. Coordination is achieved
through the definition of standard behaviors that are known to all the participants (i.e.,
common knowledge or standardization within the organization) or through explicit com-
munication.

The bounded ability of the participants also bound their ability to communicate. The
changing environment and the available technology may limit the ability to use standard
behaviors throughout the organization. Under the basic rationality assumption the partic-
ipants will attempt to find the best (or sufficient) way to operate within the organization
and the given constraints. Such rationality leads to the grouping of participants, emergence
of control structures, and development of roles and rules.

Transaction Costs Analysis

Transaction cost analysis emphasizes the opportunistic nature of the participants in orga-
nizations. It views organizations as a mechanism for optimizing the cost of transactions
between the players in the domain. Organizations are viewed as means of optimizing the
control of the achievement of goals as well as the means towards achieving the goals. This
analysis demonstrates the benefits of using social structures. In particular it demonstrates
how under some circumstances social structures improve the efficiency in which a group
of agents can achieve goals. The underlying assumptions of the transaction cost analysis
approach can be stated as follows [146]:

1. markets and firms are alternative mechanisms for completing a related set of trans-
actions;
2. whether a set of transactions ought to be executed across markets (between organi-
zations) or within an organizations depends on the relative cost of using each mode;
3. the costs of establishing and executing complex agreements across a market vary with
the characteristics of participants who are involved with the transaction on the one
hand, and the objective properties of the environment on the other; and
4. transactions between and within organizations are affected (and constrained) by the same human and transaction factors (although such factors may appear in different manifestations).

It is argued that market arrangements fail due to the characteristics of the participants and of the environment. These two factors are combined in two ways. The first is the combination of the limited cognitive abilities of the participants and the high uncertainty in the environment [144, 148].

This causes the failure of the market because market arrangements depend on the ability of the participants to reach agreements as to the details of future transactions. Environments that display high levels of uncertainty reduce the ability of the participants (due to their bounded cognitive abilities) to predict anything but the short term future. Although long term agreements are required in order to minimize the cost, in an uncertain environment the participants are unable to reach such agreements with respect to future transactions with specific outcomes.

In environments with high uncertainty the failure of the market gives way to the establishment of long term agreements with relatively general specifications as to future transactions, e.g., employment agreements. Together with these agreements appropriate monitoring and control mechanisms are also established. A set of such agreements with the corresponding control mechanisms is the basis for the establishment of an organization.

The second factor is the combination of the opportunistic nature of the participants and the small-numbers condition. The small-numbers condition occurs when there are only a small number of participants with the capability to achieve a desired goal. This maybe because of natural monopolies, the history of the interaction, etc. [144].

Again the failure of the market gives way to the establishment of organizations with long term agreements and specialized auditing mechanisms. Short term opportunism is replaced by long term reward systems.

Although under certain conditions markets give way to the establishment of organizations, such organizations are also affected (and constrained) by the characteristics of the participants. In particular the bounded abilities of the participants combined with the opportunistic nature of the participants affect the optimal size of the organization [147].

Control mechanisms are developed by having some participants in the organization adopting control (i.e., management) roles. The bounded ability of these participant limits the number of participants that they can control. This is due to the large number of decisions that should be made, amount of information to be processed, and limitations on communication facilities. This limit is referred to as the span of control. As the nature of the inter-organization agreements is fairly general and participants have an opportunistic nature there exists a distortion in the compliance with organizational goals (as communicated from one control level to the other). The level to which participants comply with the organizational goals is referred to as the compliance parameter. With multiple control levels the effect of the compliance parameter is accumulative.

The limit imposed by the span of control forces a growing organization to increase the number of levels in its control. As the number of control levels increase so does the
3.2. RELATED APPROACHES

CHAPTER 3. REQUIRED FEATURES AND RELEVANT APPROACHES

damaging effect of the compliance parameter. This affects the cost of the transaction (i.e., the achievement of the goal) that the organization has engaged in performing. When reaching such a size organizations will tend to engage in market agreements, e.g., subcontracting.

The major strength of the transaction cost analysis approach is its formal and rigid formulation of organizations and markets. This has also been the main criticism, coming mainly from researchers adopting the natural systems approach. They argue that there are many other factors associated with the participants that are not factored into the appropriate equations and formulae.

There are two reasons why the transaction cost analysis is relevant to the work presented in the following chapters. The first is that the it adopts assumptions similar to the assumptions adopted here. In particular it ignores the informal factors that are part of human behavior.

As to the second reason, there are a number of approaches in DAI that focus on market like approaches to the coordination of a group of agents [59, 115]. The transaction cost analysis provides a clear motivation to the use of social structures as means of improving the performance of a group of agents under some conditions.

Contingency Theory and the Levels Model

Contingency theory emphasizes the contingencies that are inherent in the environment. It is argued that organizations develop in order to minimize the effects of unpredictable and uncontrolled parameters. Under this approach there is a distinction between the rationality associated with the technology (i.e., the means for achieving the goals) and the rationality associated with the organization achieving its goals. Rationality with respect to the technology is referred to as technical rationality and rationality with respect to the organization is referred to as organizational rationality [121].

Organizational rationality is the primary focus of contingency theory. According to Thompson it is some result of:

1. environmental constraints that the organization must face;
2. contingencies in the environment that the organization must meet; and
3. variables that the organization can control.

Under norms of organizational rationality, organizations try to protect their core technologies from environmental influences. This is achieved by surrounding the core technology with input and output components. Such “buffering” components may take a variety of forms, e.g., stockpiling, preventive maintenance on equipment, indoctrination of the participants, maintaining product warehouses, etc. It is also inherent in the adopted organizational structure.

4 Technical rationality is similar to rationality as defined by Simon [110].
In highly unstable environments organizations operating under norms of organizational rationality will attempt to smooth input and output as much as possible and to anticipate and adapt to changes that can not be buffered or smoothed. If all else fails the organization will have to prioritize the goals to be achieved and ration the allocation of its resources (i.e., participants and technologies) to the goals according to their priority.

The model of organizational structure as suggested in contingency theory builds upon a functional analysis of organizations performed by Parsons [80]. According to this analysis organizational structures are composed of three levels that are distinguished by their functionality. The first is a technical level focused on achieving the organizational goals. The second is a managerial level focused on controlling the technical level by ensuring that the necessary resources are available through planning and design. The third is an institutional level focused on the position of the organization in the environment, determining the domain, and establishing its boundaries.

According to contingency theory since the organization is embedded in the environment the structure (as suggested by Parsons) is an outcome of the need of the organization to protect the core technology. The technology level is associated with the core technology, the institutional levels with the interfacing to environment, and the managerial level with mediating between the two. Since the focus of each component is different they would naturally adopt different operational modes. The technical level will adopt a closed rational systems model (and the corresponding technical rationality), the managerial level will adopt a natural systems model, and the institutional level will adopt an open systems model.

There are two reasons why this approach is relevant to the work presented in the following chapters. This first reason is the interesting distinction that the model makes between technical and organizational rationality. Both the technology and the structure are regarded as the means for achieving the organizational goals. The technical rationality refers to the way a particular technology is selected from the available. The organizational rationality refers to the way a particular structure is selected from the available structures. In the model of artificial organizations presented in the following chapters we make a similar distinction. We allow the organization to reason separately about the choice of technology and about the choice of structure.

The second reason comes from the clear need identified by the contingency theory and the levels model for reasoning at the institutional level about the organization as a whole. The model of artificial organizations described in the following chapters recognizes this need. It provides a framework in which reasoning about the organization as a whole, including its goals and adopted structure, is facilitated.

### 3.2.3 Command, Control, and Communication

One of the more popular models of organizations used for describing military organizations is the Command, Control, and Communication (C3) model [56, 64, 65, 73]. As mentioned earlier military organizations perform critical tasks. They thus strictly adhere to formal organizational structures. Furthermore they use training and discipline in an attempt to
eliminate all informal aspects associated with natural human behavior. Although military organizations are probably one of the oldest forms of formal organizations in human society, their formal structures have changed over the centuries.

Military organizations and their structures have been documented and investigated by many historians, sociologists, psychologists, and others. Organization Theory has been focusing on the definition and investigation of human organizations. Such definitions include the different types of relationships that exist within such organizations. Although there are many forms of military organizations the relationships between the various components are typically relatively formal and limited in their type. The C3 model of organizations is to some extent a simplified model of the relationships between various components of the same organization.

As the name suggests, C3 models define organizational structures in terms of the command, control, and communication relationships that exist between the various units of a military organization. Despite their relative simplicity there is still no single, clear, and agreed C3 model. As noted in the conclusions of a three-day symposium on command and control organized by the US Office of the Secretary of Defence [149]: “there is no adequate foundation for a theory of command and control, and hence no guiding principles for system design and evaluation.”

It is important to note here the distinction between a C3 organizational model and a C3 system. A C3 organizational model is used to specify the relationships between the components of the organization. A C3 system is a particular technology that is used in support of an adopted C3 model. It is used to facilitate the interactions between the components of the organization, e.g., conveying commands, establishing communication links, generating control instructions, etc.

There are many definitions of command, control, and communication relationships [46, 95, 149]. Some of these definitions look both at the nature of the C3 relationships as well as the role of a commander, controller, and the underlying communication system. Nevertheless there is general agreement that in some form or another:

**Command** primarily involves the authority and responsibility to determine the objectives of the organization and the sub-units of the organization;

**Control** primarily involves the authority and responsibility for specification and modification of the detailed plans required for achieving the objectives and the monitoring of the execution of these plans; and

**Communications** primarily involves the sharing of information with respect to the state of the environment, the state of the organization, the state of the achievement of objectives, and the state of the execution of plans.

An organization can be described as a set of units or individuals with various command, control, and communication relationships between them. These relationships are with respect to different objectives, types of plans, and communication protocols. The
specialization of the units of such an organizations comes both from their knowledge and capabilities and their internal and external relationship.

The knowledge and capability is expressed in terms of the types of objectives a unit can achieve, its training and knowledge of different standard operating procedures, and its mastering of various communication protocols. The internal relationships are expressed in terms of the C3 relationships between the various sub-units. The external relationships indicate the level of autonomy and responsibility the unit has.

The C3 models are relevant to the work described in the following chapters because of the type of organizations they model. C3 models are primarily designed to model organizations that operate under real-time constraints and under mission critical circumstances. As such they share the assumptions we make here about modelling real-time embedded distributed systems.

### 3.2.4 Bounded Rationality

Bounded rationality emphasizes the assumption of limited ability of the participants to make rational decisions when the amount of information and the uncertainty in the environment are large. It suggests that organizations play a critical role in reducing the amount of information that is considered by dividing the labor and introducing roles and reducing the number of options available to the decision maker by standardizing the decision processes and introducing rules [76, 108].

The notion of rationality is directed to the process of decision making rather than to the actual decisions made. It is referred to as procedural rationality [109]. The participants can make decisions given the roles and rules that they adopt and under the expectation that other participants with whom they interact will behave according to roles and rules that they adopt [105].

Roles are defined as prescription for the decision making processes that govern the behavior of the participant playing the role. Simon [108] refers to these prescriptions as composed of decision premises and provides the following definition:

> A role – in terms of this definition – is a social prescription of some, but not all, of the premises that enter into an individual’s choices of behavior.”

H. A. Simon, Administrative Behavior [108]

In general it is assumed that these premises are with respect to a generation function – a function that generates alternative courses of action, a value function – a function that evaluates the possible alternatives, and a selection function – a function that selects a course of action given the alternatives and their respective values [106]. As an example of a selection function, it is argued by Simon [76, 104] that most humans tend to select a satisficing solution rather than an optimal solution.

---

5 March and Simon originally refer to these rules as programs to emphasize their somewhat formal nature.
Rules are specification of programmed or routinized behaviors and are environment dependent. They are defined as a sequence of steps describing the activities of a participant [24, 76]. Each such rule is applicable for a given situation and the participant can be required to use different rules for different situations. Furthermore some of the steps in the rules may depend on particular aspects of the situation, e.g., the price of a product.

The rules can be hierarchical in the sense that some of the steps may be other rules, or sub-processes. The steps can also involve observation of some other activity or of the environment. March and Simon [76] identify that the rules (or the programs in their terminology) can specify the production of outputs given the appropriate inputs. Further, when rules are considered in relation to other participants or rules, they specify the coordination activity. Such coordination can be with respect to the activities of the participants or with respect to the outputs of the rules.

Recall that open rational systems assume that the changing environment affects the organization and in particular the choices of the technology available. Under certain circumstances the environment may change in a way that was anticipated before and the generation function can not generate any alternatives. Alternatively it may change in a way that none of the alternatives are evaluated as satisfactory. When this is a recurrent situation there is a need for the development of new roles or rules. It is recognized that such situation require the process of problem solving or innovation [76, 105].

Note that in this work we consider organizations of artificial agents. Our current understanding of how to automatically solve problems and invent new approaches is relatively limited. Our understanding of how to perform such tasks under limited time and resources is even smaller. In the following chapters we do not further address these problems. We assume that all possible structures and technology required will be available to the artificial organizations. If the artificial organization will be faced with a situation in which the known structures and technology are not satisfactory then the organization will fail in achieving its goals.

Recall that contingency theory and the levels model identify two types of rationality: (1) technical rationality; and (2) organizational rationality. Simon and March have primarily focused on aspects of bounded rationality that relate to the technical aspects, that is to the process of technically achieving the goal of the organization.

One can apply similar principals and discuss bounded rationality with respect to the organizational aspects. That is, the bounded ability of the organization to consider and adopt social structures. Adopting a social structure involves modifications to the mental attitudes of the participants. Selecting the best structure as well as modifying the mental attitudes of the participants requires resources.

If we assume that the organization (including its participants) have limited resources then it can not always consider the best social structure or continuously adopt structures. We would refer to the us the bounded organizational rationality of an organization.
3.2.5 Philosophical Models

There have been a number of approaches to the definition of a “group” of agents in philosophy and social science literature [11, 50, 137]. Naturally such work has primarily focused on human agents. Tuomela [135] conducts an analysis of the different approaches to the definition of a “social group”.

Tuomela’s Social Groups

In his earlier work Tuomela has primarily focused on notions of group intentions, group actions, and mutual beliefs [132, 136]. Recent work includes discussions and definitions of group goals and a detailed discussion on the definition of a social group [134, 135]. Tuomela provides the following definition of a social group [135] (pp. 192–193):

“Collective G is a social group in the core sense if and only if:

- G has a socially existing authority-system;
- Its [i.e., G’s] members, assumed to mutually believe that G is for them “we”;
- Are, collectively taken, sufficiently motivated to participate in the use of the (or a) socially existing authority-system so that the group will can be mutually believed by them to result;
- There is a mutual belief in G to the effect that its members are motivated to participate in the use of the (or a) believed authority-system; and
- The motivation in these two cases is in part due to the respective mutual belief.”

The above definition is based on the terms group will, mutual belief, and socially existing authority-system. The term group will is based on the individual wills of the members of the group. The individual will is meant to represent the force behind the intentions of the individual agent. That is the nature of the agent that forces it to actually achieve the goals (or execute the actions) it intends to achieve (or execute).

A mutual belief of a set of agents G in a proposition φ is defined to be that every member of G believes that φ and also that G mutually believes φ. This is a self referential definition similar to that provided by us in previous work [69].

A set of agents G have an authority-system if and only if there is a transformation function $f_a$ which describes the process of G’s members forming a group will on the basis of their individual wills. The set of agents have a socially existing authority-system if $f_a$ is mutually believed by G. In simple terms the above definition states that a social group is a set of agents that:

1. mutually believe that they are a social group;
2. have a mutually believed procedure that enables them to adopt joint intentions;
3. are all willing to use this procedure; and

4. mutually believe that the group is willing to use this procedure.

The above definition seems to imply that a social group has no existence independent of the existence of its members. Indeed Tuomela [133, 135] argues that “there are no (holistic or other) social properties”. He suggest that social predicates do exist and states that “a statement coached in holistic social terminology will be true in a vicarious sense if and only if there are some individualistic entities which make it true (satisfy it)”.

This approach is relevant to our work because it provides philosophical justifications for the definitions of social groups and in particular the joint mental attitudes of such a group. Here we are interested in ascribing mental attitudes to the components of a distributed system. Tuomela’s work provides us with an inspiration.

Tuomela’s approach is based on the assumption that the agent is the basic social unit. Other social entities are composed of basic social units that are associated by some set of interrelations. These assumptions come from the focus of Tuomela on groups of humans. Given that we are dealing with software systems we are not bound by these constraints.

3.2.6 Formal Mathematical Models

One of the methods of specifying a computer system is to use formal methods [139]. In such an approach a mathematical notation is used as a specification language. The use of a mathematical notation allows us to be precise in the way we define the computer system.

The desired behavior of the system depends on the meaning we ascribe to the specification language, that is, its semantics. It follows that in order to be able to prove that the behavior of the computer system will follow the desired behavior we must provide a semantic model for our language and axiom system. Providing a semantic model for a language and axiom system entails two steps: (1) describing the semantic model; and (2) proving that it is a model of the language and axiom system.

Proving that a particular semantics is a model of a axiom system is done by proving soundness and completeness. An axiom system $\mathcal{S}$ is sound with respect to a semantic model $\mathcal{M}$ if every formula provable from $\mathcal{S}$ is valid with respect to $\mathcal{M}$. $\mathcal{S}$ is complete with respect to $\mathcal{M}$ if every formula that is valid with respect to $\mathcal{M}$ is provable from $\mathcal{S}$.

The combination of a formal specification language and a corresponding semantic model allows us to investigate different types of system behaviors. This can be done either by proving different mathematical statements or by validating different behaviors in the semantic model.

Given the above requirements we are primarily looking for a mathematical model that will allow us to represent: (1) distributed systems with concurrent activity; and (2) the different mental attitudes of the agents and groups of agents that are required for exhibiting bounded rational behavior. In addition we require that: (3) there is a constructive decision procedure for the chosen model.
For the first requirement, Computational Tree Logics (CTL) [36] have been used over the past few years for representing concurrent distributed systems. For the second requirement, modal logic have been used over the past few years for representing the mental state of agents. In particular the Belief-Desire-Intention (BDI) logic has been used in the development of commercial agent-oriented systems [29]. In addition, the BDI logic (as presented by Rao and Georgeff [90]) already incorporates the CTL logic.

It is important to note that the BDI logic only relates to the mental attitudes of Belief, Desires, and Intentions. These attitudes are only with respect to a particular state of the world. That is, having a belief that this is the current state, having a desire to bring about this state, and have an intention towards bringing about this state with particular means. The logic does not include any aspect of the relationships that may exist between agents. In particular, the logic does not include any facilities for explicitly specifying the ability of an agent to affect the mental attitudes of another agent, the ability of an agent to control the decisions taken by another agent, or the commitment that one agent has to communicate its beliefs to another agent.

As for the third requirement, Halpern and Moses [54] have provided a complete analysis of constructive decision procedures for modal logics of knowledge. Rao and Georgeff [90] have extended this decision procedure to a BDI logic that includes reference to CTL [36].

It seems then that an obvious candidate to use as the first step in the development of an organization-oriented model is the BDI logic. In the following section we provide a brief overview of how this is done.

Note that we are considering distributed systems that operate in a dynamic and uncertain environment. That is, not only does the environment change while the system is executing but the system only has limited knowledge about this environment. To allow for this limited knowledge and the uncertainty about this knowledge in our model one could use possible worlds. These possible worlds represent the possible multiple worlds in which the system can be. The limited knowledge is represented as an accessibility relation over these worlds. This accessibility relation represents the worlds that are indistinguishable by the agent.

Furthermore, we are considering distributed systems that have multiple courses of actions available to them and that select the next course of action to follow. In addition following or executing a particular course of action may not always succeed. One can consider a successful execution as separate from a failed execution. To allow for both the choice and the possible outcomes in our model one can use a branching future in the Computational Tree Logic.

### 3.2.7 Approaches to Developing Multi-Agent Systems

Early approaches to the development of multi-agent systems have primarily focused on the development of advanced agents that operate in an environment with multiple agents. In particular they focused on the basic social ability of the single agent. Such approaches include the work of Rao and Georgeff [88], early work of Wooldridge and Jennings [63, 151], and early work by Singh and Asher [113, 114].
Recall that the social behavior of the multi-agent system includes the cooperation and coordination mechanisms. Also recall that the social structure of a distributed computer system includes the assignment of responsibilities to the components and the identification of the various interactions between them. Most approaches to the development of multi-agent systems do not represent the social behavior or structure explicitly. Rather, they are “hard-wired” into the behavior of the individual agents.

One can identify two approaches to cooperation and coordination. These approaches are related to the adopted model of the agent. The first approach considers action as an outcome of an explicit cognitive process. Cooperation and coordination are established through the coordination of the mental attitudes of the agents in the group. In this approach a role can be viewed as a set relationships (or constraints) between the mental attitudes of a member of the organization and the other members. Organizational structures are represented as collections of such roles. We refer to this approach as the cognitive approach to organizational structures.

The second approach considers action as an outcome of a commitment made by the agent to performing such actions when required. Cooperation and coordination are established through the allocation of actions (or commitments) made by the agents. In this approach a role can be viewed as a collection of actions (or commitments to perform such actions). Short term commitments lead to a market like organization and long term commitment lead to the establishment of persistent organizations. We refer to this approach as the task-allocation approach to organizational structures.

In both of the above approaches the developer of the system determines the social behavior and structure that is required and then “translates” it into the required individual behavior of each of the agents. The agents are then developed according to these requirements. Note that the design process moves from abstract notions such as social behavior and structure to low level descriptions of individual behavior (i.e., from top to bottom). The final specification is of the low level details of single agent behavior.

In the resulting system the abstract notions of social behavior and structure emerge from the behavior of the individual agents. These abstract notions exist only in the eye of the beholder. They are not part of the resulting system. We refer to this approach as the “bottom-up approach” to the development of multi-agent systems.

In the bottom-up approach there is no way to explicitly specify the behavior of the distributed system as a whole or the behavior of any of its sub-systems without explicitly specifying the behavior of the single agents. Where as the approaches described in the following chapters use a form of abstraction suitable for such specifications.

Cognitive Approach to Organizational Structures

The cognitive approach requires an explicit representation of the mental attitudes of the group of agents. Furthermore it is primarily concerned with mechanisms for coordinating these mental attitudes.

BGI approaches, to the development of multi-agent systems, that include formalization of joint mental attitudes have been recently proposed by a number of researchers. BGI
joint mental attitudes are an extension to the BGI model of a single agent. They are an abstract specification of the Beliefs, Goals, and Intentions of a set (or set of sets, or sets of sets... etc.) of agents. These are referred to as Mutual Beliefs, Joint Goals, and Joint Intentions respectively. Examples include the work of Kinny et al. [69], Tambe [118], and Singh [111]. Note that the joint mental attitudes are still based on the definition of an agent as the basic unit.

The plans that are used by these sets of agents (referred to as Joint or Team Plans) include a high-level specification of the required coordination. This higher level of abstraction provides the developer with a way of explicitly describing the required coordination. Furthermore it allows the set of agents to dynamically modify the joint plan but still maintain coordination. Examples of work that define team plans are Kinny et al. [69], Grosz and Kraus [52], and Tambe [117, 119].

Tambe [117] has also conducted simulations with sets of artificial agents that use shared team plans (i.e., as common knowledge). Tambe has demonstrated that under real-time and bounded rationality constraints the hypothesis of the Open Rational Systems is correct. That is the use of standardization (i.e., shared coordination methods) has reduced communication and improved the overall performance of the set of agents.

Recently there have also been some preliminary work on defining roles and organizational structures for sets of agents. Again the primary focus is on the agent as the basic unit. Examples include the work of Tidhar [123], Cavedon and Sonenberg [17], Collinot et. al. [22], Singh [112], and Yu [154].

Task-Allocation Approach to Organizational Structures

The task-allocation approach requires an explicit representation of the assignment (or allocation) of goals (or tasks) to agents. Furthermore, it is primarily concerned with mechanisms for assigning goals (or allocating tasks) and identifying interactions between the assigned goals.

Examples include early work by Durfee, Lesser, and Corkill [33, 35] and later work by Decker, Lesser, Garvey and others [25, 28, 47]. In their model they use two primary types of information to coordinate the activities of a group of agents: (1) a task relationship structure; and (2) a global plan for the group of agents.

A task relationship structure represents the various types of relationships that exist between different tasks. These include the sub-task, temporal precedence, and many other relationships. These relationships only depend on the tasks and the domain and not on the agents that have been allocated the tasks.

A global plan for a set of agents includes a task allocation and instructions on the coordination of the execution of the tasks. It is generated at run time and is based on the task relationship structure, the abilities of the agents, and the current task allocation. The coordination instructions include a schedule for the achievement of tasks. This schedule ensures that constraints imposed by the task relationship structure are adhered to in the activities of the autonomous agents.
The work presented here is a combination of the cognitive and task-allocation approaches to organizational structures. It provides a generalization of many of the ideas on multi-agent systems described above. We prescribe to the cognitive approach to actions. Cooperation and coordination are established through special social mental attitudes and joint plans. The social mental attitudes are a relationship between two organizations (or agents) and a particular state of the world. A role is viewed as a set of social mental attitudes between a given organization (or agent) and any other organization and any state of the world. A joint plan specifies the coordination of the activities performed by the various members of an organization.

One can require that a social mental attitude enforce a constraint over the joint mental attitudes of the two respective organizations (or agents). This will result in an approach similar to the cognitive approach. In this case one would primarily focus on the relationships between the two organizations (or agents) specified in the social mental attitudes.

Alternatively one can require that a social mental attitude and joint plan enforce a commitment to bring about the specified state of the world. The social mental attitude can then be adopted by the respective organizations for a short term, resulting in a short term dynamic task allocation, or over a long term, resulting in a more long term stable task allocation. In this case one would primarily focus on the relationship between the first organization (or agent) and the state of the world specified in the social mental attitude.

This combined approach to representing organizational structures is developed incrementally. We provide a hierarchy of models that include models of a single agent in a single agent world, a single agent in a multi-agent world, team-oriented models, and organization-oriented models.

3.2.8 Other Related Approaches

Computer Models and Organization Theory

Using computer models for simulation of human organizations has been an active area since the mid 1950's. Work in this field is concerned with providing researchers in OMT with computer models for simulation and analysis of the behavior of human organizations [53]. Such models have taken four main forms [20]: (1) descriptive simulations; (2) analytic simulations; (3) design simulations; and (4) training simulations.

Descriptive simulations are mainly used for modelling existing organizations and providing researchers with tools for explaining why human organizations behave in the manner they do.

Analytic simulations are mainly used for modelling surreal organizational models. Such models provide researchers with tools for analyzing the effects of particular assumption on organizational behavior.
3.2. REQUIRED FEATURES AND RELEVANT APPROACHES

Design simulations are mainly used for determining the suitability of different types of organizations to particular goals or environments. Such models provide researchers with the tools for modelling the effectiveness of existing organizations and to predict the behavior of organization under new circumstances.

Training simulations are mainly used for training humans to function better within organizations. Such models allow for initiation of staff into new organizations and positions and training of staff when the organization changes.

It seems that most simulation models have adopted two main approaches for modelling an organization: (1) a macro model in which the participants are simplified; and (2) a micro model in which the simulation focuses on the reasoning within a limited number of individuals. In the former approach the cognitive aspects of human reasoning are represented using some statistical or compact model. Such approach focus on the behavior of the organization as a whole. In the latter approach, the cognitive aspects of the participants are fairly elaborate but the organizational model and reasoning is simplified.

As computer models of human reasoning and distributed systems have advanced researchers in OMT have started using such models. Such recent work has combined computer simulation techniques with organization theory models. The field is called Computational Organization Theory [10].

The work described in the following chapters includes both an advanced model of the participants of an organization as well as an advanced model of organizational structures. Although such models are provided under very limiting assumptions they could be useful for use in computational organization theory. In particular for analytic and design simulations.

Distributed Object-Oriented Software Engineering

A field of software engineering that considers problems similar to those described here is “Distributed Object-Oriented Software”. This field is primarily concerned with extending and applying the object-oriented approach to the development of distributed systems. In particular in the development of analysis and design methodologies that support the development of distributed systems.

Examples include the work of Seinturier and Duchien [98, 99] and Batory and O’Malley [4]. The relevance of such work to the work described here is primarily as a basis for the development of a methodology for developing organization-oriented systems.

Enterprise Modelling

The work on enterprise modelling has focused on the development of ontologies for representing human enterprises. The motivation behind such models is that they be included in information systems used by the enterprise to improve the performance and usability
of such systems [40, 43]. One aspect of human enterprises is the organizational structures adopted.

The organization-oriented model described in the following chapters can be used as one approach to enterprise modelling. One can view it as another ontology for representing human organizations under particular assumptions.

3.3 Limitations of Related Approaches

The main limitation of the BGI model is that the behavior of the agent does not include a facility for an un-reasoned response, e.g., a reflex reaction. The BGI agent can not act if it has not adopted a goal and a corresponding intention. This excludes the development of models in which the developer would require the system to ignore the reasoning process and react with some predetermined action. An example could be the shutting down of the equipment being controlled in case of a major failure independent of the state of the equipment or the environment.

The main limitation of the naive approach to creating intentions is that typically the number of possibilities would be very large and the agent would have limited resources. In particular a real-time system would have limited time to react.

The main limitation of the pre-planned approach to creating intentions is that it may reduce the survivability of the organization, that is, its ability to respond to new or unknown circumstances. This is because the organization only has a limited repertoire of plans to select from. It may be the case that the organization will not have the plan to achieve its objective under the new circumstances.\(^6\)

The main limitation of the bottom-up approach is that when the environment changes it is very difficult for the multi-agent system to dynamically modify its social behavior. Recall that the social behavior is considered as a means the distributed system employs to achieve its objectives. Given that there is no explicit representation of the social behavior it is difficult to reason about the best social behavior (i.e., the best means) for achieving the objectives under the new circumstances. Furthermore, it is difficult to clearly identify the relationship between the adopted social behavior and the adopted individual behavior.

An example in which these limitations are significant would be the coordination required between the air-conditioner, the hot-water system, and the electricity meter and the relationship between the three systems. Let us assume that both the air-conditioner and the hot water system can not operate at full capacity at the same time.

To resolve the problem of coordination we can assign the meter the responsibility to control and coordinate the whole system. Under normal circumstances, i.e., both the air-conditioner and hot-water system are on automatic mode, the meter would allocate the electricity to each of the systems. It will even ensure that the system consumes electricity when it is the cheapest. This is fine if there is no immediate need for services from one of these systems.

\(^6\)An obvious solution is that when faced with a new situation the organization could revert to the naive approach. Obviously this will result in degradation of its response time for that instance.

© Copyright Gil Tidhar, 1999 54
If the two systems are not equal and cost consideration should be ignored then one system would have to take control over the whole distributed system. An example is when the family returns from a picnic and requires a large quantity of hot water. In this case we may let the hot-water system decide on the allocation of electricity but the meter would still coordinate the operation of the air-conditioner. If it is a hot day and the air-conditioner is also required at full capacity then maybe a third approach to the social behavior could be adopted.

Although this is a simple example it highlights the need for the multi-agent system to modify its social behavior and reason about the best possible behavior for the given circumstances. The specification of these social behaviors would be used by a developer to describe the required behavior (as was done above but hopefully in a more precise way).

As to the development of a multi-agent system using the bottom-up approach, it is done at the lowest possible level. That is, the developer can not specify the behavior of the distributed system as a whole at some abstract level. Removing the abstraction and specifying the distributed system at the lowest possible level makes it very difficult to design the system and contain the complexity of the specification (see the work of Booch [8] on Abstraction and Encapsulation).

The main limitation of the multi-agent BGI approach to the problem described above is that the basic unit is still a single agent. The developer is still unable to describe the behavior of the distributed system or one of its sub-systems as a whole without explicitly specifying the complete set of agents that are part of the distributed system or sub-system.

Furthermore, the social structure adopted by the set of agents is still “hard-wired” into the behavior of the individual components. The distributed system is unable to easily modify the social structure as response to the introduction of a new objective or the changing circumstances.

The evaluation of of organization theory models described above is with respect to the problem addressed in this work. Their main limitation, in the context of this problem is that they assume, quite naturally, that the members of the distributed system (i.e., the organization) are human. As such the models have been developed to take into account the various issues that are relevant to human behavior. Such issues may not necessarily be relevant to a computer system. Examples include motivational problems, breaking formal rules, personal problems, etc.

Another limitation is the level of formality in which the models are described. As a machine, a computer system expects to be given information and instructions using a precise and formal language. The description of the model can not be in vague or general terms. Furthermore, a model of a real-time embedded distributed system that performs a mission critical role has to be validated and verified before it becomes operational. Such validation and verification can only be done with a formal specification.\footnote{It should be noted that when a mission critical computer system is developed it has to pass an extensive process of formal validation and verification. When human experts are performing the same mission critical task they only have to pass relatively narrow testing. It is assumed that the human experts would have general knowledge, experience, and common sense that will allow them to choose the right response at the required time. It is not clear if this is a valid assumption.}
Nevertheless, models of Organization and Management Theory provide an understanding of the behavior of organizations. They also provide insights into the reasons for the emergence of organizations and advantages and limitations that they bring. Such insights could be used as the basis for understanding the operations of a distributed computer system. Most importantly such models provide an inspiration for the development of multi-agent systems.

To conclude, current approaches to the specification and development of multi-agent systems have the following limitations:

**No Explicit Formal Model of Social Behavior to Support Development:** The developer of a multi-agent system is not provided with an explicit formal model of the required social behavior of the system as a whole. This forces them to manually translate the required social behavior to the individual behavior of the components.

**No Explicit Model of Social Behavior to Support Running System:** The running multi-agent system (including any of the agents) does not have an explicit model of social behavior. The system (and the agents) are unable to reason about the most suitability social behavior for achieving the current objective under the current circumstances.

**No Dynamic Modification of Social Behavior:** Without an explicit model of social behavior the running multi-agent system does not have the ability to modify its social behavior in a timely manner as a response to the changing objectives or the changing circumstances.

**No Explicit Model of the Interaction Between Social and Individual Behaviors:** The developer can not explicitly specify the required interaction between the adopted social behavior and the adopted individual behavior.

**Limited ability to Verify and Validate the Multi-Agent System:** Because there is no formal model of the social behavior it is very difficult to verify and validate the design and implementation of the multi-agent system against the required behavior.

These limitations reduce the flexibility and survivability of the multi-agent system. Furthermore they increase the time it takes to develop the system. Given that there is no complete formal model of social behavior and social structure, it is also very difficult to validate and verify the behavior of the multi-agent system. This is vital when considering mission critical systems. Most of these problems are exacerbated when considering them in the context of a real-time embedded multi-agent system with limited computational resources.

In the models described here we accept the limitation of the BGI model and the pre-planned approach but provide a formal model of a real-time embedded distributed system (including social behavior and structure) that addresses all the other limitations described above.
Chapter 4

Single Agent Systems

The model of a single agent adopted here is a Belief-Goal-Intention (BGI) model and is based on the model described by Rao and Georgeff [88]. In this chapter we present the Rao and Georgeff model of a single agent. We use this model as the basis for the development of models of multi-agent, team-oriented, and organization-oriented systems described in the following chapters.

In the BGI approach to modelling a single agent, an agent consists of a set of beliefs, goals, and intentions. Beliefs define the current state of the world as perceived by the agent, goals are the states of the world the agent would like to attain, and intentions are the agent’s adopted commitments towards attaining such goal states.

Rao and Georgeff [90] have conducted a detailed investigation of single agent models. They have also conducted investigations into the definition of rationality and decision making in the context of the BDI approach. For the purpose of completeness of this work we provide the details of the basic Rao-Georgeff model. We have made some minor modifications, primarily in notation, which we will highlight.

The description of the Rao-Georgeff model provided in this section is a direct duplication of the original work without significant enhancements or additions. For a detailed investigation of the single agent BDI model see the original Rao and Georgeff work [90].

The first modification introduced here is the use of goals rather than desires. We thus have a model of Beliefs, Goals, and Intentions. From a philosophical perspective the difference between goals and desires is mainly with respect to two issues: (1) different desires of the agent or team may be inconsistent; and (2) the agent or team may have desires which it has no means of achieving. The consistent achievable desires are referred to as goals.

As mentioned above, one of the attributes of the behavior of the agent or team, which we consider as rational, is that they do not hold inconsistent or unachievable desires. We thus restrict our discussion to agents and teams that have goals. In the remainder of this
4.1 Syntax of a Single Agent BGI System

The BGI systems we consider are extensions of Computation Tree Logics, CTL and CTL* [36, 37] that have been used extensively for reasoning about concurrent programs. We extend the branching-time logics to represent the mental state or belief-goal-intention state of an agent. These logics can then be used to reason about agents and the way in which their beliefs, goals, and actions can bring about the satisfaction of their goals.

We introduce a propositional, temporal, multi-modal logic, BGICTL. The primitives of this language include a non-empty set $U$ of primitive propositions; propositional connectives $\lor$ and $\neg$; modal operators $\text{BEL}(\text{agent believes})$, $\text{GOAL}(\text{agent has a goal})$, and $\text{INTEND}(\text{agent intends})$; and temporal operators $X$ (next), $U$ (until), $F$ (sometime in the future or eventually), $E$ (some path in the future or optionally). Other connectives and operators such as $\land$, $\supset$, $\equiv$, $G$ (all times in the future or always), $B$ (before), $A$ (all paths in the future or inevitably), can be defined in terms of the above primitives.

Modification: The work of Rao and Georgeff and the work presented here are both based on the work of Emerson [36]. Nevertheless the terms “world” and “state” are used in the work of Rao and Georgeff and the work of Emerson to mean different things. Rao and Georgeff use a set of state names as indices into each world. The valuation of a formula is done with respect to a world and a state within this world. They then associate a time point with each state. We have decided to remove any ambiguity with respect to the terms “state” and “world” as defined by Emerson. We define a world to be a temporal structure and change the truth assignment function to be with respect to a world and a time point. We can directly map this model to the state based model of Rao and Georgeff.

There are two types of well-formed formulas in these languages: state formulas (which are true in a particular world at a particular time) and path formulas (which are true in a particular world along a certain path). We inductively define the class of state formulas for BGICTL using rules S1-S4 and the class of path formulas for BGICTL using rule P0.

(S1) each atomic proposition $\phi$ is a state formula;

(S2) if $\phi$ and $\psi$ are state formulas then so are $\neg \phi$ and $\phi \land \psi$;

(S3) if $\phi$ is a path formula then $A\phi$ and $E\phi$ are state formulas;

(S4) if $\phi$ is a state formula then $\text{BEL}(\phi)$, $\text{GOAL}(\phi)$, and $\text{INTEND}(\phi)$ are state formulas; and

(P0) if $\phi$ and $\psi$ are state formulas then $X\phi$ and $\phi U \psi$ are path formulas.

Path formulas of BGICTL are restricted to be primitive linear-time temporal formulas, with no negations or disjunctions and no nesting of linear-time temporal operators. Comparing the above formation rules with those for CTL [36] one can observe that we have added the formation rule S4.

© Copyright Gil Tidhar, 1999
4.2 Modelling Air Missions with a Single-Agent System

The operators BEL, GOAL, and INTEND represent, respectively, the beliefs, goals, and intentions of the agent. Disjunctions, implications, and equivalences are defined in the classical way: \(\phi \lor \psi\) is defined as \(\neg(\neg\phi \land \neg\psi)\); \(\phi \supset \psi\) is defined as \(\neg(\phi \land \neg\psi)\); and \(\phi \equiv \psi\) is defined as \(\neg(\phi \land \neg\psi) \land \neg(\psi \land \neg\phi)\).

The language BGI\(_{CTL}\) can be used to represent the mental state of an agent, in particular its belief-goal-intention state. For example, consider an agent who has the desire to eventually win a lottery, intends to buy a lottery ticket sometime in the future, but does not believe that he will eventually win the lottery. The mental state of this agent can be represented by the following formula of BGI\(_{CTL}\): \(\text{GOAL}(\text{AF}(\text{win-lottery})) \land \text{INTEND}(\text{EF}(\text{buy-lottery-ticket})) \land \neg\text{BEL}(\text{AF}(\text{win-lottery}))\).

Following Halpern and Moses [54], we first define some basic properties of these formulae. The size of a formula \(\phi\), denoted by \(|\phi|\), is its length over the alphabet \(U \cup \{\neg, \land, (, ), \text{BEL}, \text{GOAL}, \text{INTEND}, \text{A}, \text{E}, \text{X}, \text{F}, \text{G}, \text{U}, \text{B}\}\). The depth of a formula \(\phi\), denoted by \(\text{depth}(\phi)\), is the depth of nesting of \text{BEL}, \text{GOAL}, and \text{INTEND} operators in \(U\). The formula \(\psi\) is said to be a subformula of \(\phi\), if \(\psi\) is a substring of \(\phi\). Let \(\text{Sub}(\phi)\) be the set of all subformulas of \(\phi\). Note that \(\text{depth}(\phi) < |\phi|\) and \(|\text{Sub}(\phi)| \leq |\phi|\). For example, the formula \(\phi \equiv \neg p \land \text{BEL}(p \land q) \land \text{GOAL}(q)\) has a size of 14, depth of 1, and \(|\text{Sub}(\phi)| = 7\).

4.2 Modelling Air Missions with a Single-Agent System

Using the single agent approach one has to specify what is the behavior required from each of the agents in the system. We use the analysis of the scenario provided in Section 2.2.2 as the starting point. We then proceed through a process of refinement until we get a full specification of all the agents in the system.

At the initial stage the Blue force has as sub-teams a Sector Air Defence, AEW&C aircraft, AP-3C aircraft, and ADGE. The Sector Air Defence has as sub-teams a SADC, and 2 pairs of hornets. We refer to these pairs as an Attack and a CAP respectively. The SADC is an agent and each hornet in the pair is also an agent. The AEW&C has as sub-teams an MC and two FCs. In this scenario we view the AP-3C and ADGE as agents. It follows that we need to specify the agents that will be part of the agent-oriented system:

- a single SADC agent;
- four hornet agents;
- a single MC agent;
- two FC agents;
- a single AP-3C agent; and
- a single ADGE agent.
4.2. MODELLING AIR MISSIONS WITH A SINGLE-AGENT SYSTEM

CHAPTER 4. SINGLE AGENT SYSTEMS

Given that each of the agents is totally independent we have to provide them with state formula that specifically mention agents in the environment. These are not considered agents but rather just data about the environment. Such state formula will include the state of other agents, messages from other agents, etc. It is important to note that given that there is no concept of other agents in the underlying model managing the information about other agents has to be handled by the designer rather done by the underlying system.

Note that for any state formula the designer has to specify what the state formula describes. It follows that the designer has to explicitly specify what types of beliefs can each agent have about other agents. Furthermore reasoning about the beliefs of another agent about a third agent’s beliefs becomes very cumbersome.

Also note that the agent has to store information about its beliefs about the relationships with other agents. In particular the command, control, and communication relationships. The way these beliefs affect its behavior will be “hard wired” into the agent’s plans.

If the Attack can not operate relatively autonomously then the relationships will also include control relationship between the FC and the Attack. This control will only be with respect to the choice of tactics (i.e., plans) adopted by the attack. We say that the FC and Attack pair have a “close control” relationship. This relationship will have to be implicitly modelled as one of the beliefs of the agent.

Given that different relationships can hold for different goals, the agent has to store in its database the relationships and the goals they are applicable for.

4.2.1 Plans and Coordination

We will now focus on the plans available to the 2 hornets. These plans are stored in a plan library and are included in the specification of each of the hornets.

We will first consider two plans that achieve a given goal required as part of the capabilities of a hornet agent. We will then proceed to describe the way in which the beliefs of the Command, Control, and Communication relationship affect the execution of these plans.

Plans

We consider the plan (i.e., tactics) that achieves an intercept on a given target. In particular we consider the pincer intercept. The pincer intercept involves a group of agents forming two distinct but cooperating elements in order to intercept an enemy aircraft. One element, referred to as the lead, attacks the target from the left and the other element, referred to as the wing, attacks the target from the right. The following briefly describes the stages of a pincer intercept:

- The goal to commence a pincer intercept is adopted by an agent (due to a command from a commander or as a response to the situation).
• The agent inspects its database and identifies if it is part of the lead or the wing for this particular intercept.\(^1\)

• If it is part of the lead then it performs a cutoff intercept on the target from the left.

• If it is part of the wing then it performs a cutoff intercept on the target from the right.

• Once the wing completes the cutoff intercept from the right it returns to a standard flight formation with the lead. This is done by identifying the spatial position of the lead and positioning the lead at a particular relative position and maintaining that relative position.

If the designer can ensure that both agents execute their respective plans at the same time then one would observe a coordinated behavior. It is important to note here that the agents are not interacting with each other and coordinating their activities. Rather they have been designed to react to the changes in the world (e.g., the belief about the command from the FC) in a compatible way.

Ensuring that the behavior of the agents will indeed be compatible is not a simple task. In particular if each agent can choose multiple plans to intercept the target. Note that there is no joint decision making. It follows that the designer must ensure that for all cooperating agents at any moment there is only a single (and compatible) plan that they can choose. Alternatively, if there are multiple plans the designer has to ensure that all agents will choose the same plan.

Effect of Inter-Agent Relationships

It is important to note here that the agent will have its own beliefs about the relationship between agents. Ensuring that these beliefs are consistent has to be done by the designer of the system. As noted above these relationships should be inspected by each agent as part of executing its plans.

It is clear that there is much room for confusion. This is particularly true if multiple designers are designing the system. The group of designers will have to be very closely coordinated in-order to ensure that the beliefs of all agents about the relationship have the same meaning. Furthermore, one has to ensure that these beliefs are updated appropriately across multiple agents and are taken into account in all plans.

Note again that the effect of the inter-agent relationship is “hard-wired” into the plans. It follows that the designer would be required to specify multiple plans that achieve the same objectives under different relationships. For example, in the case of close control all the agents in the attack force must communicate the possible means to achieve the intercept to the FC. This communication has to be specified explicitly by the designer in the plans. Alternatively in the tactical control no such communication is required.

---

\(^1\)Note that the lead may be a wing of another group or for another intercept at the same time.
When the relationship between the agents changes there are many coordination tasks that should be performed by the agents. These tasks include recognizing that there is a change in the relationship, ensuring that other agents also believe that there is a change, and then finally modifying the behavior of the agents in-accordance with the change. The detailed ways these tasks are performed should all be specified by the developer.

4.3 Possible-Worlds Semantics

The traditional possible-worlds semantics of beliefs considers each world to be a collection of propositions and models belief by a belief-accessibility relation \( B \) linking these worlds. A formula is said to be believed in a world if and only if it is true in all its belief-accessible worlds [54].

Cohen and Levesque [21] treat each possible world as a time-line representing a sequence of events, temporally extended infinitely into the past and the future. Formulas are evaluated with respect to a given world and an index into the course of events defining the world. The accessibility relation \( B \) is a relation between the world at an index to a set of courses of events. Intuitively, an agent believes a formula in a world at a particular index if and only if in all its belief-accessible worlds the formula is true.

We instead consider each possible world to be a tree structure with a single past and a branching future. Each tree structure denotes the optional courses of events that can be chosen by an agent in a particular world. Evaluation of formulas is with respect to a world and a time point. Hence, a time point acts as an index into a particular tree structure of the agent. The belief-accessibility relation maps a possible world at a time point to other possible worlds. The goal- and intention-accessibility relations behave in a similar fashion. More formally, we have the following definition of a Kripke structure or interpretation of a single agent system.

**Modification:** In addition to using a time tree structure we also identify the events that occur between the time points. We restrict our model to consider events that are primitive in a temporal sense. That is, they are restricted to appear between successor time points.

Given a set of time points \( T \), a relation \( \prec \) over the time points in \( T \), and two time points \( t_i, t_j \in T \), we say that \( t_j \) is a successor of \( t_i \) if and only if \( \{ t_i, t_j \} \in \prec \) and there is no \( t_k \in T \) such that \( \{ t_i, t_k \} \in \prec \) and \( \{ t_k, t_j \} \in \prec \).

We define a successor-restricted relation of \( \prec \), denoted by \( \prec^s \), to be the set of time point pairs \( \{ t_i, t_j \} \in \prec \) such that \( t_j \) is a successor of \( t_i \).

**Definition 1**

An interpretation of a single agent system (or a Kripke interpretation) is defined to be a tuple:

\[
M^{sas} = \langle \mathcal{E}, T, \prec, \mathcal{C}, \mathcal{W}, \mathcal{B}, \mathcal{G}, \mathcal{I}, \mathcal{U}, \Phi \rangle
\]

\( \mathcal{E} \) is a set of events and \( \mathcal{W} \) is a set of possible worlds. \( T \) is a set of time points and \( T_w \subseteq T \) is the set of time points in each world \( w \in \mathcal{W}. \prec \) is a total binary relation over \( T \), i.e.,
relation is dependent on the state, the mapping of is true in all the belief-accessible worlds of the agent at time

Thus the agent can change its beliefs about the options available to it.

we refer to from the corresponding attributes to the set

is defined as including a time structure rather than a state structure and hence we use

the structure

holds (or is

Georgeff [90], notably that: (1) here we refer to the relation over time points as \( \prec \) rather than \( \mathcal{R} \) and to the mapping function as \( \Phi \) rather than \( \mathcal{L} \); (2) as mentioned above a world is defined as including a time structure rather than a state structure and hence we use time points as indices into a world; and (3) we introduce the concept of events that occur between time points.

Given a time interval \( (t_i, t_j) \) and an event \( e \) that is associated with this time interval, we refer to \( t_i \) as the start time of the event and to \( t_j \) as the end time of the event.

For the remainder of this work we will at times refer to relations as if they were functions to from the corresponding attributes to the set \( \{True, False\} \). That is, for a given relation \( \mathcal{R} \) between sets \( S_1 \times \ldots S_n \) we will say that “for \( s_1 \in S_1, \ldots, s_n \) we have that \( \mathcal{R}(s_1, \ldots, s_n) \) holds (or is \( True \))” to mean that “\( (s_1, \ldots, s_n) \in \mathcal{R} \)”.

Satisfaction of formulas is denoted by \( \models \). In our model satisfaction is with respect to the structure \( M^{sas} \), a world \( w \), and a time point \( t \). The expression \( M^{sas}, w_t \models \phi \) should be read as “structure \( M^{sas} \) in world \( w \) and time point \( t \) satisfies \( \phi \)”. A path \( t_0, t_1, \ldots \), in world \( w \) is denoted by \((w_{t_0}, w_{t_1}, \ldots)\).

(S1) \( M^{sas}, w_t \models \phi \) iff \( \phi \in \Phi(w,t) \) where \( \phi \) is a primitive proposition.

(S2) \( M^{sas}, w_t \models \neg \phi \) iff \( M^{sas}, w_t \not\models \phi \).

(S3) \( M^{sas}, w_{t_0} \models \phi \) iff there exists a path \((w_{t_0}, w_{t_1}, \ldots)\) such that \( M^{sas}, (w_{t_0}, w_{t_1}, \ldots) \models \phi \).

(S4) \( M^{sas}, w_t \models \text{BEL}(\phi) \) iff \( \forall v \) satisfying \((w, t, v) \in \mathcal{B}, M^{sas}, v_t \models \phi \).

(P0) \( M^{sas}, (w_{t_0}, w_{t_1}, \ldots) \models \phi \) if \( \forall \psi \) satisfying \((w, t, \psi) \in \mathcal{G}, M^{sas}, v_t \models \phi \).

We say that an agent has a belief \( \phi \), denoted \( \text{BEL}(\phi) \), in time point \( t \) if and only if \( \phi \) is true in all the belief-accessible worlds of the agent at time \( t \). As the belief-accessibility relation is dependent on the state, the mapping of \( \mathcal{B} \) at some other state may be different. Thus the agent can change its beliefs about the options available to it.
Similar to belief-accessible worlds, for each state we also associate a set of goal-accessible worlds to represent the goals of the agent. Thus, in the same way that we treat belief, we say that the agent has a goal \( \phi \), denoted \( \text{GOAL}(\phi) \), in time point \( t \) if and only if \( \phi \) is true in all the goal-accessible worlds of the agent in time \( t \).

Intentions are similarly represented by sets of intention-accessible worlds. These worlds are ones that the agent has chosen to attempt to realize. The intention-accessibility relation \( I \) is used to map the agent’s current world and time point to all its intention-accessible worlds. We say that the agent intends a formula in a certain time if and only if it is true in all the agent’s intention-accessible worlds at that time.

Validity of formulas is defined in the standard manner, i.e., a formula is valid if it is true in every time point, in every world, in every structure. A formula \( \phi \) is said to be valid in \( M^{\text{as}} \), written as \( M^{\text{as}} \models \phi \), if \( M^{\text{as}}, w, t \models \phi \) for every world \( w \in W \) and every time point \( t \in T_w \). Similarly, one can define validity and satisfiability with respect to a class \( M^{\text{as}} \) of structures. We say that \( \phi \) is valid with respect to a class \( M^{\text{as}} \) of structures, written as \( M^{\text{as}} \models \phi \), if \( \phi \) is valid in all structures in \( M^{\text{as}} \), and say that \( \phi \) is satisfiable with respect to a class \( M^{\text{as}} \) of structures if \( \phi \) is satisfiable in some structure in \( M^{\text{as}} \).

We adopt standard definitions of a relation being total, serial, transitive, and euclidean. More formally, we have:

(Total) \( \forall w \forall s \exists t (s, t) \in \prec_w \);

(Serial) \( \forall w \forall s \exists v (w, s, v) \in B \);

(Transitive) \( \forall w, v, x \forall s \text{ if } (w, s, v) \in B \text{ and } (v, s, x) \in B \text{ then } (w, s, x) \in B \);

(Euclidean) \( \forall w, v, x \forall s \text{ if } (w, s, v) \in B \text{ and } (w, s, x) \in B \text{ then } (v, s, x) \in B \).

We consider the class of structures \( M^{\text{as}} \) which requires \( \prec \) to be total and does not impose any constraints on the accessibility-relations \( B, G, \) and \( I \).

4.4 Basic Axiom System

In this section, we discuss a basic axiom system for our BGI\(_{\text{CTL}}\). As CTL is contained within BGI\(_{\text{CTL}}\), the axiomatization of BGI\(_{\text{CTL}}\) will contain all the CTL axioms and inference rules. For completeness we include the full-set of axioms and inference rules for CTL as given by Emerson [36] here.

4.4.1 Axiomatization of CTL Component

(CTL1) All validities of propositional logic;

(CTL2) \( EF\phi \equiv E(\text{True } U \phi) \);

(CTL2b) \( AG\phi \equiv \neg EF\neg \phi \);
(CTL3) \( \text{AF}\phi \equiv \text{A}(\text{True} \cup \phi) \);

(CTL3b) \( \text{EG}\phi \equiv \neg \text{AF}\neg \phi \);

(CTL4) \( \text{EX}(\phi \lor \psi) \equiv \text{EX}\phi \lor \text{EX}\psi \);

(CTL5) \( \text{AX}\phi \equiv \neg \text{EX}\neg \phi \);

(CTL6) \( \text{E}(\phi \cup \psi) \equiv \psi \lor (\phi \land \text{EX}(\phi \cup \psi)) \);

(CTL7) \( \text{A}(\phi \cup \psi) \equiv \psi \lor (\phi \land \text{AX}(\phi \cup \psi)) \);

(CTL8) \( \text{EX}\text{True} \land \text{AX}\text{True} \);

(CTL9) \( \text{AG}(\xi \supset (\neg \psi \land \text{EX}\xi)) \supset (\xi \supset \neg \text{A}(\phi \cup \psi)) \);

(CTL9b) \( \text{AG}(\xi \supset (\neg \psi \land \text{EX}\xi)) \supset (\xi \supset \neg \text{AF}\psi) \);

(CTL10) \( \text{AG}(\xi \supset (\neg \psi \land (\phi \supset \text{AX}\xi))) \supset (\xi \supset \neg \text{E}(\phi \cup \psi)) \);

(CTL10b) \( \text{AG}(\xi \supset (\neg \psi \land (\phi \supset \text{AX}\xi))) \supset (\xi \supset \neg \text{EF}\psi) \);

(CTL11) \( \text{AG}(\phi \supset \psi) \supset (\text{EX}\phi \supset \text{EX}\psi) \);

(CTL-Gen) If \( \vdash \phi \) then \( \vdash \text{AG}\phi \);

(MP) If \( \vdash \phi \) and \( \vdash \phi \supset \psi \) then \( \vdash \psi \).

4.4.2 Axiomatization of BGI component

In addition to the CTL-component, we will adopt the K-axiom of modal logic, for beliefs, desires, and intentions. The K-axiom is the minimal system for normal modal logics. This axiom states that if an agent believes \( \phi \) and believes that \( \phi \supset \psi \) then he will believe \( \psi \). We extend this constraint to desires and intentions.

(B-K) \( \text{BEL}(\phi) \land \text{BEL}(\phi \supset \psi) \supset \text{BEL}(\psi) \);

(G-K) \( \text{GOAL}(\phi) \land \text{GOAL}(\phi \supset \psi) \supset \text{GOAL}(\psi) \);

(I-K) \( \text{INTEND}(\phi) \land \text{INTEND}(\phi \supset \psi) \supset \text{INTEND}(\psi) \).

We also have the generalization rule for beliefs, goals, and intentions, which states that any valid formula is believed, is a goal, and is intended.

(B-Gen) If \( \vdash \phi \) then \( \vdash \text{BEL}(\phi) \);

(G-Gen) If \( \vdash \phi \) then \( \vdash \text{GOAL}(\phi) \);

(I-Gen) If \( \vdash \phi \) then \( \vdash \text{INTEND}(\phi) \).
4.5 Soundness and Completeness

An axiom system \( S \) is sound with respect to a class \( \mathcal{M} \) of structures if every formula provable from \( S \) is valid with respect to \( \mathcal{M} \). \( S \) is complete with respect to \( \mathcal{M} \) if every formula that is valid with respect to \( \mathcal{M} \) is provable from \( S \).

To show the soundness and completeness of the BGI\textsuperscript{sas} CTL system we the small model property for our logic. This property states that if a formula is satisfiable, then it is satisfiable in a ‘small’ finite model, where ‘small’ is interpreted as a size that is bounded by some function, say \( f \), of the length of the input formula [36]. An equivalence relation of small finite index can be defined on states which collapse a possibly infinite model to a small finite model. Such a construction is called the quotient construction.

In modal logics of knowledge and belief this construction is used to generate a model, called the canonical model. Soundness and completeness of an axiom system can be shown with respect to this canonical model. In other words, a formula that is provable in the modal system can be shown to be satisfiable in the canonical model and vice versa. To take into account the CTL component included in the BGI\textsuperscript{sas} CTL system we have to create a Hintikka structure [36].

To show that a formula \( \phi_0 \) has the small model property we first show that it has an infinite tree model with finite branching bounded by the size of the formula. We then apply the quotient construction to this infinite tree model to obtain a finite pseudo-Hintikka structure of size exponential (at most) to the length of the formula. From this pseudo-Hintikka structure we can unwind a finite model of size exponential (at most) to the length of the formula. This establishes the finite model property for our logic.

We now formalize the above description by giving precise definitions for the concepts introduced above and then formally prove the small model theorem for the BGI logic. This then leads us to a more constructive decision procedure for checking satisfiability of formulas.
4.6 Small Model Theorem

We assume that the formula for which we are checking satisfiability, \( \phi_0 \), is in positive normal form. A formula \( \phi \) can be transformed into a positive normal form formula by pushing negations inward as far as possible using the propositional equivalences (i.e., \( \neg(\phi \land \psi) \equiv \neg\phi \lor \neg\psi \); \( \neg(\phi \lor \psi) \equiv \neg\phi \land \neg\psi \)) and temporal equivalences (i.e., \( \neg\text{AG}\phi \equiv \text{EF}\neg\phi \); \( \neg(\phi \lor \psi) \equiv \text{E}(\neg\phi \lor \psi) \)). This results in propositions and belief, desire, and intentions modal operators being negated. The positive normal form of the formula \( \neg\phi_0 \) is denoted by \( \neg\phi_0 \).

A positive normal formula of the form \( \text{E}\phi \) is called an optional formula or O-formula and is denoted by \( \gamma \); similarly a positive normal formula of the form \( \text{A}\phi \) is called an inevitable formula or I-formula and is denoted by \( \delta \).

The closure of \( \phi_0 \), denoted by \( cl(\phi_0) \), is the least set of subformulas which satisfy the following conditions:

- if \( \psi \in \text{Sub}(\phi_0) \) then \( \psi \in cl(\phi_0) \);
- if \( \text{EF}\psi, \text{EG}\psi, \text{E}(\phi \lor \psi) \) or \( \text{E}(\phi \land \psi) \) \( \in cl(\phi_0) \) then \( \text{EX}\text{E}\text{F}\psi, \text{EX}\text{E}\text{G}\psi, \text{EX}(\phi \lor \psi) \) or \( \text{EX}(\phi \land \psi) \in cl(\phi_0) \), respectively;
- if \( \text{A}\text{F}\psi, \text{A}\text{G}\psi, \text{A}(\phi \lor \psi) \) or \( \text{A}(\phi \land \psi) \in cl(\phi_0) \) then \( \text{AX}\text{A}\text{F}\psi, \text{AX}\text{A}\text{G}\psi, \text{AX}(\phi \lor \psi) \) or \( \text{AX}(\phi \land \psi) \in cl(\phi_0) \), respectively.

The extended closure of \( \phi_0 \) is defined as: \( \text{ecl}(\phi_0) = cl(\phi_0) \cup \{ \neg\phi : \phi \in cl(\phi_0) \} \).

We define an elementary formula to be a formula with one of the following forms: \( \phi, \neg\phi, \text{EX}\phi, \text{AX}\phi, \text{BEL}\phi, \neg\text{BEL}\phi, \text{GOAL}\phi, \neg\text{GOAL}\phi, \text{INTEND}\phi \) or \( \neg\text{INTEND}\phi \). Any formula that is not an elementary formula will be called a non-elementary formula. Each non-elementary formula is classified as either a conjunctive formula \( \alpha \equiv \alpha_1 \land \alpha_2 \) or a disjunctive formula \( \beta \equiv \beta_1 \lor \beta_2 \). Clearly, \( \phi \land \psi \) is an \( \alpha \) formula and \( \phi \lor \psi \) is a \( \beta \) formula. The fix-point characterizations of temporal formulas are used to classify them as \( \alpha \) or \( \beta \) formulas. For example, \( \text{AF}\phi \equiv \phi \lor \text{AXAF}\phi \) is a \( \beta \)-formula and \( \neg\text{AF}\phi \equiv \neg\phi \land \neg\text{AXAF}\phi \) is an \( \alpha \) formula. Table 4.1 shows the \( \alpha \) and \( \beta \) rules for \( \text{BGI}^{\text{sas}}_{\text{CTL}} \).

<table>
<thead>
<tr>
<th>( \alpha )</th>
<th>( \alpha_1 )</th>
<th>( \alpha_2 )</th>
<th>( \beta )</th>
<th>( \beta_1 )</th>
<th>( \beta_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \phi \land \psi )</td>
<td>( \phi \land \psi )</td>
<td>( \psi )</td>
<td>( \phi \lor \psi )</td>
<td>( \phi \lor \psi )</td>
<td>( \psi )</td>
</tr>
<tr>
<td>( \text{A}(\phi \land \psi) )</td>
<td>( \neg\psi )</td>
<td>( \phi \lor \text{AXA}(\phi \land \psi) )</td>
<td>( \text{A}(\phi \lor \psi) )</td>
<td>( \text{A}(\phi \lor \psi) )</td>
<td>( \psi )</td>
</tr>
<tr>
<td>( \text{E}(\phi \land \psi) )</td>
<td>( \neg\psi )</td>
<td>( \phi \lor \text{EXE}(\phi \land \psi) )</td>
<td>( \text{E}(\phi \lor \psi) )</td>
<td>( \text{E}(\phi \lor \psi) )</td>
<td>( \psi )</td>
</tr>
<tr>
<td>( \text{AG}\psi )</td>
<td>( \psi )</td>
<td>( \text{AXAG}\psi )</td>
<td>( \text{AX}\psi )</td>
<td>( \text{AXAF}\psi )</td>
<td>( \text{AXAF}\psi )</td>
</tr>
<tr>
<td>( \text{EG}\psi )</td>
<td>( \psi )</td>
<td>( \text{EXEG}\psi )</td>
<td>( \text{EF}\psi )</td>
<td>( \text{EF}\psi )</td>
<td>( \text{EXEF}\psi )</td>
</tr>
</tbody>
</table>

Table 4.1: Alpha and Beta Rules for \( \text{BGI}^{\text{sas}}_{\text{CTL}} \)

Similar to CTL-logics [36], we define a pre-structure \( \text{sas}^{\Phi} \) to be a tuple \( \langle \mathcal{E}, T, \prec, \mathcal{C}, W, B, \mathcal{G}, \mathcal{T}, U, \Phi \rangle \) except that the binary relation \( \prec_w \) for each world \( w \) is not
required to be total. An interior node of a pre-structure is one with at least one successor. A frontier node is one with no successors. A fragment is a pre-structure whose graph is a directed acyclic graph such that all of its nodes satisfy PC0-2, LC0, BC0, DC0, and IC0, and all of its interior nodes satisfy LC1, BC1, DC1, and IC1 as defined below.

Propositional Consistency Rules:

(PC0) if $\neg \phi \in \Phi(w,t)$ then $\phi \notin \Phi(w,t)$;

(PC1) if $\alpha \in \Phi(w,t)$ then $\alpha_1 \in \Phi(w,t)$ and $\alpha_2 \in \Phi(w,t)$;

(PC2) if $\beta \in \Phi(w,t)$ then $\beta_1 \in \Phi(w,t)$ or $\beta_2 \in \Phi(w,t)$.

Local Consistency Rules:

(LC0) if $AX \phi \in \Phi(w,t)$ then for all successors $s$ of $t$, $\phi \in \Phi(w, t)$;

(LC1) if $EX \phi \in \Phi(w,t)$ then for some successor $s$ of $t$, $\phi \in \Phi(w, t)$;

Basic BGI-Consistency Rules:

(BC0) if $BEL(\phi) \in \Phi(w,t)$ and $(w, t, v) \in B$ then $\phi \in \Phi(v, t)$;

(BC1) if $\neg BEL(\phi) \in \Phi(w,t)$ then $\exists v$ such that $(w, t, v) \in B$ and $\neg \phi \in \Phi(v, t)$;

(GC0) if $GOAL(\phi) \in \Phi(w,t)$ and $(w, t, v) \in G$ then $\phi \in \Phi(v, t)$;

(GC1) if $\neg GOAL(\phi) \in \Phi(w,t)$ then $\exists v$ such that $(w, t, v) \in G$ and $\neg \phi \in \Phi(v, t)$;

(IC0) if $INTEND(\phi) \in \Phi(w,t)$ and $(w, t, v) \in I$ then $\phi \in \Phi(v, t)$;

(IC1) if $\neg INTEND(\phi) \in \Phi(w,t)$ then $\exists v$ such that $(w, t, v) \in I$ and $\neg \phi \in \Phi(v, t)$;

A set of formulas $F$ that satisfies all of the PC0-PC2 rules will be called a propositional CTL tableau. Following Halpern and Moses [55] we say that a set $F$ of formulas is fully expanded if for every formula $\phi \in F$ and subformula $\psi$ of $\phi$, either $\psi \in F$ or $\neg \psi \in F$. A fully-expanded propositional CTL tableau is one that is both a propositional CTL tableau and is fully-expanded.

All non-elementary formulas are marked in a fully-expanded propositional CTL tableau. A node whose label is a fully-expanded propositional CTL tableau is called a world-time-point.

Definition 2

A $BGIP_{sas}\text{ CTL-tableau}$ (for $\phi_0$), $M_{sas}$, is a tuple $\langle E, T, \prec, C, W, B, G, I, U, \Phi \rangle$ (with $\phi_0 \in \Phi(w, t)$ for some $w \in W$ and some $t \in T_w \in T$) which meets the following conditions:

- the propositional consistency rules (PC0-2);
- the local consistency rules (LC0-2);
• each eventuality is fulfilled; and
• the basic BGI-consistency rules BC0, BC1, GC0, GC1, IC0, and IC1 are satisfied.

The primary differences between the above definition and that of CTL-logics are the basic BGI-consistency rules which define the constraints on the belief-, goal-, and intention-accessibility relations. These constraints correspond to the K-axiom system for normal modal logics [61].

If $M^{sas}$ is a $BGI^{sas}_{CTL}$-tableau, then for each world $w$ and time point $t$ of $M^{sas}$ and each eventuality $\xi$ in $ecl(\phi_0)$ such that $M^{sas}, w_t \models \xi$, there is a fragment, $\text{DAG}[w_t, \xi]$, which certifies the fulfilment of $\xi$ in world $w$ at time point $t$ in $M^{sas}$. If $\xi$ is of the form $\text{AF} \psi$, then $\text{DAG}[w_t, \xi]$ can be obtained by taking time point $t$ and all time points along all paths in $w$ emanating from $t$ up to and including the first time where $\psi$ is true. This fulfilling fragment is said to be cleanly embedded in $M^{sas}$ [55].

Now we would like to apply the following quotient construction to collapse equivalent world-time-points.

**Definition 3**

Let $M^{sas} = \langle \mathcal{E}, T, \prec, \mathcal{C}, W, \mathcal{B}, \mathcal{G}, \mathcal{T}, U, \Phi \rangle$ be a model of $\phi_0$, let $H$ be a set of formulas, and let $\equiv_H$ be an equivalence relation on $W$ and $T$ induced by agreement on the formulas in $H$, i.e., $w_t \equiv_H v_s$ whenever $\forall \psi \in H, M^{sas}, w_t \models \psi$ iff $M^{sas}, v_s \models \psi$. We use $[w_t]$ to denote the equivalence class $\{v_s \equiv_H w_t\}$ of $w_t$. Then the quotient structure of $M$ by $\equiv_H$ $M/\equiv_H$ is $\langle \mathcal{E}^q, T^q, \prec^q, \mathcal{C}^q, W^q, \mathcal{B}^q, \mathcal{G}^q, \mathcal{T}^q, U, \Phi^q \rangle$ where $G^q = T^q = \{[w_t] : t \in T_w \text{ and } w \in W\}; \prec^q = \{([w_t]_1, [w_s]_1) : (t, s) \in \prec_w\}; \mathcal{B}^q = \{([w_t]_1, [v_t]_1) : \text{BEL}^-([w_t]) \subseteq [v_t]\}$; similarly for $\mathcal{G}^q$, and $\mathcal{T}^q; \Phi^q([w_t]) = \Phi(w_t) \cap H$. Normally, $H$ is taken to be $ecl(\phi_0)$. $\text{BEL}^- (X) = \{\phi : \text{BEL} (\phi) \in X\}$ and similarly for $\text{GOAL}^- (X)$ and $\text{INTEND}^- (X)$.

Unlike normal modal logics where the above quotient construction will result in a model (called the canonical model), the above quotient construction for $BGI^{sas}_{CTL}$ may not result in a model. This is because cycles are introduced in the fulfilling fragments and these fragments are no longer cleanly embedded, but are just contained in $M^{sas}$. However, the construction still yields useful information which can be unwound into a proper model. Hence, we have the following definition of a pseudo-$BGI^{sas}_{CTL}$-tableau.

**Definition 4**

A pseudo-$BGI^{sas}_{CTL}$-tableau (for $\phi_0$) is a structure $M = (W, T, \prec, \Phi)$ (with $\phi_0 \in \Phi(w, t)$ for some $w \in W$ and some $t \in T_w \in T$) which meets the following conditions:

1. the propositional consistency rules (PC0-2);
2. the local consistency rules (LC0-2); and
3. each eventuality is pseudo-fulfilled in the following sense:
4.6. SMALL MODEL THEOREM  

CHAPTER 4. SINGLE AGENT SYSTEMS

(a) $\forall F \psi \in \Phi(w, t)$ (respectively, $A(\phi \cup \psi) \in \Phi(w, t)$) implies there is a finite fragment, called $\text{DAG}[w, t, \forall F \psi]$ (respectively, $\text{DAG}[w, t, A(\phi \cup \psi)]$), rooted at world $w$ and time point $t$ contained in $M^{sas}$ such that for all frontier nodes $s$ of the fragment, $\psi \in \Phi(w, s)$ (respectively and for all interior nodes $u$ of the fragment, $\phi \in \Phi(w, u)$);

(b) $\exists F \psi \in \Phi(w, t)$ (respectively, $A(\phi \cup \psi) \in \Phi(w, t)$) implies there is a finite fragment, called $\text{DAG}[w, t, \exists F \psi]$ (respectively, $\text{DAG}[w, t, A(\phi \cup \psi)]$), rooted at world $w$ and time point $t$ contained in $M^{sas}$ such that for some frontier node $s$ of the fragment, $\psi \in \Phi(w, s)$ (respectively and for all interior nodes $u$ of the fragment, $\phi \in \Phi(w, u)$);

4. the basic BGI-consistency rules BC0, BC1, DC0, DC1, IC0, and IC1 are satisfied.

Theorem 1  

Let $\phi_0$ be a $BGI^{sas}_{CTL}$ formula of length $n$. Then we have the following equivalences:

1. $\phi_0$ is $BGI^{sas}_{CTL}$-satisfiable;
2. $\phi_0$ has a model $M^{sas}$ with finite branching in each world bounded by $O(n)$;
3. $\phi_0$ has a finite pseudo-$BGI^{sas}_{CTL}$-tableau of size $\leq \exp(n)$;
4. $\phi_0$ has a finite model $M^{sas}$ of size $\leq \exp(n)$.

Proof:

We show that (1) $\rightarrow$ (2) $\rightarrow$ (3) $\rightarrow$ (4) $\rightarrow$ (1).

(1) $\rightarrow$ (2): Suppose, $M^{sas}$ $w_t \models \phi_0$. Using the standard mechanisms of CTL [36] one can unwind the world $w$ into an infinite tree with finite branching bounded by $O(n)$ all of whose eventualities are fulfilled. Some of these nodes will be accessible to other worlds through the $B$, $G$, and $I$-accessibility relations. Each one of these worlds can be unwound into an infinite tree with finite branching, as before. This process is carried out recursively until there are no embedded modal formulas. This process will terminate due to the finite size of the formula $\phi_0$.

(2) $\rightarrow$ (3): Let $M^{sas}$ be a class of structures, with $M^{sas}$ in $M^{sas}$, such that $M^{sas}$, $w_t \models \phi_0$. We show that the quotient structure $M' = M^{sas} /\equiv_{ctd(\phi_0)}$ is a pseudo-$BGI^{sas}_{CTL}$-tableau.

The proof that the quotient structure $M'$ satisfies the propositional consistency rules PC0-PC2 and the local consistency rules LC0-LC1 is trivial. Now we show that $M'$ pseudo-fulfills each eventuality and also satisfies the BGI-consistency rules.

We consider the pseudo-fulfilment of $\forall F \psi$; the other cases are similar. In the original structure $M^{sas}$, $\forall F \psi$ is true and hence there must exist a finite fragment $\text{DAG}[w_t, \forall F \psi]$ with root $t$ in $w$ cleanly embedded in $M^{sas}$. However, the quotient construction introduces
cycles into such fragments. Therefore, to obtain a fragment in the quotient structure which is acyclic we copy the original fragment and remove all duplicate labels. Given two time points \( t \) and \( s \) in \( w \), we let the deeper state replace the shallower. After removing all such duplicates we have a finite fragment DAG \( \psi' \) that is contained in the quotient structure \( M' \).

We show that the quotient construction satisfies (BC0). The other conditions can be proved in a similar manner. Let \( \text{BEL}(\psi) \in \Phi(w, t) \) and \( (w, t, v) \in B \). From the quotient construction we have \( M', [w_t] \models \text{BEL}(\psi) \) and \( ([w_t], [v_t]) \in B' \). From the definition of beliefs we have \( M^{sas}, [v_t] \models \psi \). As a result \( \psi \in \Phi(v, t) \).

(3) \( \rightarrow \) (4): The only difference between a pseudo-BGI\textsuperscript{sas}\text{CTL} tableau for \( \phi_0 \) and a BGI\textsuperscript{sas}\text{CTL} tableau for \( \phi_0 \) is the pseudo-fulfilment as opposed to the fulfilment of eventualities. One can follow a procedure similar to the one used in CTL-logics [36] (Page 1034-1036) to splice together copies of the DAG’s, one for each eventuality in each state, to obtain a BGI\textsuperscript{sas}\text{CTL} tableau model for \( \phi_0 \).

The size of the model can be shown to consist of \( m \times N^2 \) nodes, where \( m \) is the number of eventualities and \( N \) is the number of nodes in the model. The number of nodes \( N \leq 2^n \), where \( n \) is the length of \( \phi_0 \). In other words the size of the model is \( \exp(n) \).

(4) \( \rightarrow \) (1): This follows directly from the definition.

There are two major differences between the small model theorem for CTL [36] and the one given above. First, instead of a single branching tree structure we have multiple branching tree structures, one for each world. Second, there are non-temporal modal operators for beliefs, desires, and intentions that define accessibilities across the multiple trees. As we will see later, various constraints on these accessibility relations lead to different classes of models.

The small model theorem for normal modal logics, such as the modal logic for belief [55], is relatively straightforward as it does not have the complications introduced by the temporal operators.

Having proved the finite model theorem we know that we can construct a finite model for checking the satisfiability of a formula in BGI\textsuperscript{sas}\text{CTL}. In the next section we provide an algorithm for constructing a pseudo-BGI\textsuperscript{sas}\text{CTL} tableau and then extracting a BGI\textsuperscript{sas}\text{CTL} tableau from it.

### 4.7 Algorithm

The algorithm for constructing a pseudo-BGI\textsuperscript{sas}\text{CTL} tableau consists of five different procedures. The first procedure expands a set of formulas to a propositional CTL tableau. The second procedure expands a propositional CTL tableau into a fully expanded propositional CTL tableau.

A formula \( \phi \) is said to be a witness [55] if it does not satisfy one of the PC0-PC2 rules or \( \phi \) is a subformula of \( \psi \) and neither \( \phi \) nor \( \lnot \phi \) is in the label. If all formulas are ordered
according to their length, then a least witness is a witness with the least length. When there are more than one least witness, a witness is arbitrarily chosen to expand a tableau. When a witness has been expanded, it is marked as having been expanded. Starting from the given formula \( \phi_0 \) as the root of the tableau, one can choose the least witness one after the other until the tableau is a fully-expanded propositional CTL tableau.

The only unmarked formulas in a fully-expanded propositional CTL tableau are elementary formulas, i.e., formulas of the form \( \phi, \neg\phi, \text{EX}\phi, \text{AX}\phi, \text{M}(\phi), \neg\text{M}(\phi) \) (where \( \text{M} \) is one of \( \text{BEL}, \text{GOAL}, \text{or INTEND} \)). The third and fourth procedures independently expand the elementary formulas of CTL (i.e., \( \text{EX}\phi \) and \( \text{AX}\phi \)) and elementary formulas of BGI (i.e., \( \text{M}(\phi) \) and \( \neg\text{M}(\phi) \)), respectively. The former results in the creation of \( \prec \)-successors ensuring the satisfaction of local consistency rules \( \text{LC0} \) and \( \text{LC1} \) and the latter results in the creation of \( \mathcal{B}, \mathcal{G}, \) and \( \mathcal{I} \)-successors ensuring the satisfaction of the BGI-consistency rules \( \text{BC0}, \text{BC1}, \text{DC0}, \text{DC1}, \text{IC0}, \) and \( \text{IC1} \).

The fifth procedure checks for satisfiability of labels. Any label that is blatantly inconsistent, i.e., contains \( \phi \) and \( \neg\phi \) for some formula \( \phi \), is unsatisfiable and the corresponding node is not marked as being ‘satisfiable’. Depending on a label being a fully-expanded CTL tableau or not, different satisfaction conditions apply for the label. When the root node of a pseudo-BGI\( ^{\text{sas}} \)CTL-tableau is marked ‘satisfiable’ we can say that the label of such a node is BGI\( ^{\text{sas}} \)CTL-satisfiable.

Algorithm 1: Algorithm for Constructing a Pseudo-BGI\( ^{\text{sas}} \)CTL-Tableau

Begin:

- **Step 1:** Construct a tree consisting of a single node \( w_0 \), with \( \Phi(w_0) = \{\phi_0\} \).
- **Step 2:** Repeat until none of (a) – (d) below applies:

  (a) *Forming a propositional CTL tableau:* If node \( w_i \) is a leaf of the tree, \( \Phi(w_i) \) is not blatantly inconsistent, \( \Phi(w_i) \) is not a propositional CTL tableau, and \( \phi \) is the least witness to this fact, then:

    1. if \( \phi \) is an \( \alpha \)-formula, create a son of this node, labelled by \( \Phi(w_i) \cup \{\alpha_1, \alpha_2\} \) and mark \( \phi \) as ‘expanded’.
    2. if \( \phi \) is a \( \beta \)-formula, create two sons of this node, labelled by \( \Phi(w_i) \cup \{\beta_1\} \) and \( \Phi(w_i) \cup \{\beta_2\} \), respectively, and mark \( \phi \) in the label of each son as ‘expanded’.

  (b) *Forming a fully expanded propositional CTL tableau:* If node \( w_i \) is a leaf of the tree, \( \Phi(w_i) \) is not blatantly inconsistent, \( \Phi(w_i) \) is not a fully expanded propositional CTL tableau, and \( \phi \) is the least witness to this fact, then create two sons labelled by \( \Phi(w_i) \cup \{\phi\} \) and \( \Phi(w_i) \cup \{\neg\phi\} \), respectively.

  (c) *Expanding elementary CTL formula:* If node \( w_i \) is a leaf of the tree, \( \Phi(w_i) \) is not blatantly inconsistent, \( \Phi(w_i) \) is a fully expanded propositional CTL tableau, all
4.7. ALGORITHM

CHAPTER 4. SINGLE AGENT SYSTEMS

non-elementary CTL formulas at the node are marked ‘expanded’, and \( \Phi(w_i) \) is labelled with formulas \( AX\phi_1, \ldots, AX\phi_m, EX\psi_1, \ldots, EX\psi_k \), then create \( k \) \( \prec \)-successors of node \( w_i \), labelled with the set \( \{\phi_1, \ldots, \phi_m, \psi_1\} \ldots, \{\phi_1, \ldots, \phi_m, \psi_k\} \), respectively. If there is an ancestor with an identical label the edge is directed to the existing ancestral node.

(d) Expanding elementary BGI formula: If node \( w_i \) is a leaf of the tree, \( \Phi(w_i) \) is not blatantly inconsistent, and \( \Phi(w_i) \) is a fully expanded propositional tableau, then

1. if \( \Phi(w_i) \) contains \( \neg B\phi \ldots \neg B\phi_m \) then create \( m \) \( B \)-successors of node \( w_i \), labelled with \( B(\Phi(w_i)) \cup \{\neg \phi_j\} \), where \( 1 \leq j \leq m \);
2. if \( \Phi(w_i) \) contains \( \neg G\phi \ldots \neg G\phi_m \) then create \( m \) \( G \)-successors of node \( w_i \), labelled with \( G(\Phi(w_i)) \cup \{\neg \phi_j\} \); where \( 1 \leq j \leq m \);
3. if \( \Phi(w_i) \) contains \( \neg I\phi \ldots \neg I\phi_m \) then create \( m \) \( I \)-successors of node \( w_i \), labelled with \( I(\Phi(w_i)) \cup \{\neg \phi_j\} \); where \( 1 \leq j \leq m \).

(e) Marking nodes ‘satisfiable’: If node \( w_i \) is not marked ‘satisfiable’ then mark node \( w_i \) satisfiable if one of the following conditions holds:

1. node \( w_i \) is not a fully-expanded propositional CTL tableau and one of its sons is marked satisfiable;
2. node \( w_i \) is a fully-expanded propositional CTL tableau such that for every eventuality \( \phi \in \Phi(w_i) \) there exists a fragment DAG\[w_i, \phi\] rooted at node \( w_i \) contained in the tableau which certifies pseudo-fulfilment of \( \phi \), and all \( \prec \)-, \( B \)-, \( G \)-, \( I \)-successors of node \( w_i \) are marked ‘satisfiable’;
3. node \( w_i \) is a fully expanded propositional CTL tableau, there are no eventuality formulas or formulas of the form \( \neg B\phi \), \( \neg G\phi \), \( \neg I\phi \) in \( \Phi(w_i) \), and \( \Phi(w_i) \) is not blatantly inconsistent.

• **Step 3:** If the root of the tree is marked ‘satisfiable’, then return ‘\( \phi_0 \) is satisfiable’; otherwise return ‘\( \phi_0 \) is unsatisfiable’.

**End.**

The construction so far has given us a pseudo-BGI\(sas_{\text{CTL}}\)-tableau for the satisfaction of the formula \( \phi_0 \). We need to extract a model for \( \phi_0 \) from this. In particular, we need to collapse the nodes where the non-elementary formulas are being expanded or fully-expanded and take the nodes which are world-time-points, i.e., fully-expanded propositional CTL tableau. Intuitively, an edge labelled with \( \prec \) between two ‘satisfiable’ world-time-points \( w_i \) and \( w_j \) corresponds to \( (i, j) \in \prec_w \). Similarly, an edge labelled with \( B \) between two world-time-points \( w_i \) and \( v_i \) corresponds to \( (w, i, v) \in \mathcal{B} \), where \( w \) is the world-state \( w_i \) and \( v \) is the world-state \( v_i \). Similarly, we can extract the nodes for desires and intentions.

More formally, a world-time-point \( v \) is said to be an \( \prec \)-successor to a world-time-point \( w \) if and only if there is a path in \( w_0, \ldots, w_k \) in the pseudo-BGI\(sas_{\text{CTL}}\)-tableau such that \( w_0 \)
= w and w_k = v, the edge from w_0 and w_1 is labelled with ≺ and for all j with 0 < j < k, w_j is an internal node and w_{j+1} is a successor of w_j in the pseudo-BGI^{sas}_CTL-tableau. Similarly, we can define a world-time-point v to be a $\mathcal{B}$successor (respectively, $\mathcal{G}$ or $\mathcal{I}$) of world-time-point w.

4.8 Completeness

We are now in a position to prove the soundness and completeness of BGI^{sas}_CTL-system using tableaus. In other words, we can prove that a formula $\phi_0$ is BGI^{sas}_CTL-provable if and only if $\phi_0$ is marked ‘satisfiable’ in a pseudo-BGI^{sas}_CTL-tableau.

To prove this we show that if a node $w_j$ is not marked ‘satisfiable’ then $\phi_{w_j}$ is inconsistent or $\neg \phi_{w_j}$ is BGI^{sas}_CTL-provable, where $\phi_{w_j}$ is the conjunction of formulas in the label of node $w_j$. By induction on the depth of the BGI^{sas}_CTL-tableau we then show that if the root node $w_0$ is not marked ‘satisfiable’ then $\neg \phi_0$ is BGI^{sas}_CTL-provable. We prove this by establishing the following two lemmas. The first lemma shows that if the label of a node $w_j$ is inconsistent then the label of its $\prec$ predecessor is also inconsistent. The second lemma proves a similar result for $\mathcal{B}$ predecessors. The proof for these two lemmas can be found in the work of Rao and Georgeff [90].

**Lemma 1** If $\phi_{w_j}$ is inconsistent and $(w_i, w_j) \in \prec$ as constructed in the pseudo-BGI^{sas}_CTL-tableau then $\phi_{w_i}$ is inconsistent, where $\phi_{w_i}$ and $\phi_{w_j}$ are the conjunction of propositions in nodes $w_i$ and $w_j$, respectively.

**Lemma 2** If $\phi_j$ is inconsistent and $(w_i, w_j) \in \mathcal{B}$ as constructed in the pseudo-BGI^{sas}_CTL-tableau then $\phi_i$ is inconsistent, where $\phi_i$ and $\phi_j$ are conjunctions of propositions in node $w_i$ and $w_j$, respectively.

**Theorem 2** The BGI^{sas}_CTL-system is a sound and complete axiomatization with respect to $M^{sas}$.

**Proof:**

Proving the soundness of the BGI^{sas}_CTL-system is straightforward. We sketch the completeness of the BGI^{sas}_CTL-system.

Suppose $\phi_0$ is valid. Then $\neg \phi_0$ is unsatisfiable. We apply the above tableau-based decision procedure to $\neg \phi_0$. All nodes whose label includes $\neg \phi_0$ will not be marked ‘satisfiable’.

We now show that if a node $w_i$ is not marked ‘satisfiable’ then $\vdash \neg \phi_i$ or $\phi_i$ is inconsistent. We proceed by induction on the height of a node $w_i$ (i.e., the length of the longest path from $w_i$ to a leaf of the pre-tableau).

**Case 1:** Node $w_i$ is a leaf of the tree.

From Step 2 (e) the node is not marked ‘satisfiable’ if and only if $\Phi(w_i)$ is blatantly inconsistent. Hence, $\phi_i$ is inconsistent.
4.8. COMPLETENESS  

CHAPTER 4. SINGLE AGENT SYSTEMS

**Case 2:** Node $w_i$ is an internal node of the tree and is not a fully expanded propositional CTL tableau.

From Step 2 (e) the node $w_i$ is not marked ‘satisfiable’ if and only if none of the successors of $w_i$ are marked ‘satisfiable’. The successors of $w_i$ must have been created using either the $\alpha$ rule or the $\beta$ rule. If $w_{i_1}$ and $w_{i_2}$ are the successors of $w_i$, then by our induction hypothesis both $w_{i_1}$ and $w_{i_2}$ are not marked ‘satisfiable’. Therefore, $\phi_{i_1}$ and $\phi_{i_2}$ are inconsistent or $\vdash \neg\phi_{i_1}$ and $\vdash \neg\phi_{i_2}$. By propositional reasoning we can show that $\vdash \neg\phi_i \land \neg\phi_{i_2} \rightarrow \neg\phi_i$. Hence, we have $\vdash \neg\phi_i$. Similarly, if an $\alpha$ rule is used, we can show that $\vdash \neg\phi_i$.

**Case 3:** Node $w_i$ is an internal node of the tree and is a fully expanded propositional CTL tableau.

From Step 2 (e) the node $w_i$ is not marked ‘satisfiable’ if any one of the following conditions apply:

1. a $\mathcal{B}$, $\mathcal{G}$, $\mathcal{I}$-successor is not marked ‘satisfiable’;
2. an $\prec$-successor is not marked ‘satisfiable’;
3. an eventuality formula is not fulfilled.

If node $w_j$ is a $\mathcal{B}$-successor of node $w_i$ and node $w_j$ is not marked ‘satisfiable’ then it follows that $\phi_j$ is inconsistent. From Lemma 2 we can conclude that $\phi_i$ is inconsistent and hence will not be marked ‘satisfiable’. Similar, arguments hold for $\mathcal{G}$, and $\mathcal{I}$-successors of node $w_i$.

If node $w_j$ is an $\prec$-successor of node $w_i$ and node $w_j$ is not marked ‘satisfiable’ then it follows that $\phi_j$ is inconsistent. From Lemma 1 we can conclude that $\phi_i$ is inconsistent and hence will not be marked ‘satisfiable’.

The proof of node $w_i$ not being marked ‘satisfiable’ when $\mathsf{EF}\psi$ is in $w_i$, but $\mathsf{EF}\psi$ not being fulfilled, is identical to the corresponding proof for proving the completeness of the CTL system [36]. The proofs of $\mathsf{AF}\psi$, $\mathsf{E}(\phi \ U \psi)$, $\mathsf{A}(\phi \ U \psi)$ can be carried out likewise.

■
Chapter 5

Multi-Agent Systems

In this chapter we use the BGI model of the single agent described above to create a BGI model of a multi-agent system. The multi-agent model extends the notions of belief, goals, and intentions for a single agent to include a reference to the agent holding these mental attitudes. That is, not only do these attitudes exist for separate agents, but they are also recognized as such. This recognition is both in the model of the multi-agent system and the model of each agent in that system. This allows us to specify an agent reasoning about the beliefs, goals, and intentions of other agents. In particular we can specify an agent that explicitly reasons about the coordination with other agents. We adopt an approach similar to that described by Halpern and Moses [54] for creating a model of a multi-agent system.

5.1 Syntax

The syntax of multi-agent systems described here is similar to the syntax described by us in previous work [69]. We restrict the discussion only to single agents. In the following chapter we will extend our model to include teams.

Again similar to CTL, there are two types of formulas in the logic: state formulas (these are evaluated at a given time point in a given world) and path formulas (these are evaluated along a given path in a given world). For multi-agent systems we only need to modify the definition of a state formula with respect to a belief, goal, and intention as follows:

- if $\phi$ is state formula and $\alpha$ is an agent then $\text{BEL}(\alpha, \phi)$, $\text{GOAL}(\alpha, \phi)$ and $\text{INTEND}(\alpha, \phi)$ are state formulas;

Intuitively, the operators $\text{BEL}$, $\text{GOAL}$, and $\text{INTEND}$ represent, respectively, the beliefs, goals, and intentions of the respective agent.
5.2 Possible-Worlds Semantics for Multi-Agent Systems: Informal Discussion

In this section, we informally discuss a possible-worlds semantics for a multi-agent system and the mental attitudes of multiple agents. Let us first consider the different elements that should be considered for a model of a multi-agent system. We will proceed to discuss the model adopted here and some motivation for this selection. In the following section we provide the formal semantic model for a multi-agent system.

A multi-agent system is composed of a set of agents, a particular universe of discourse (i.e., the domain), and information about this universe. Each agent is uniquely associated with an agent constant (i.e., the agent name). The agents operate in the context of time and a world. Changes in the multi-agent system happen through the occurrence of events.

5.2.1 The Domain

The universe of discourse (i.e., the domain) is represented as a set of first-order syntactical entities. The information about the domain is represented as relations between these entities. The relationships between the entities may change as an outcome of the changes to the multi-agent system.

5.2.2 A Model of Events

Changes to the world are brought about through the occurrence of events. Each event is an action taken by one of the agents in the multi-agent system. We consider the existence of a special agent named environment that causes environmental changes.

The attempt of an agent to act may succeed, in which case the multi-agent system will change in two ways: (1) the effects of the event will change the multi-agent system; and (2) a successful attempt will be noted. Alternatively the attempt may fail, in which case the only change will be the noting of a failed attempt.\(^1\)

We associate with each event two different time points that represent the start and end of the event. We require the end point to be a successor of the start point (i.e., there is no time point that follows the start and precedes the end). Given that there are multiple agents that can cause different events concurrently there may be multiple events that share the same start and end points.

Using the above model concurrency is modelled by nondeterminism and fairness [36, 37]. The semantics of the concurrent performance of events by multiple agents is given by a computation tree. In such a tree the concurrent performance of events is modelled by branches that represent all possible nondeterministic sequences of these actions. By

\(^1\)In this work we are not concerned with the frame problem and for simplicity we ignore this problem. That is the success of an event will result in the world changing in exactly the way specified by the event. No other changes will occur and we do not consider any side effects. On the other hand a failed attempt does not have any side effects and the only change to the world is the progress of time.
choosing a sufficiently fine level of granularity of the events any behavior that can be produced by a truly concurrent multi-agent system can be modelled by the interleaving model.

In addition to the notion of nondeterminism we also need to assume *fair scheduling* (or fairness). In the context of concurrent systems fairness represents the fact the different agents may operate at different speeds. That is, in the same time period a faster agent may perform more events than a slower agent. Here we model concurrency as a set of sequences of events that are connected in the starting point. This results in a computation tree structure. To reflect the different speeds of the agents we should ensure that in each of the possible sequences the number of events performed by each agent is proportionate to its relative speed. For example, consider a system with two agents one twice as fast as the other. A fair representation of this system will have in each of the possible sequence of events that the number of events performed by the faster agent is twice the number of events performed by the slower agent.

### 5.2.3 A Model of Time

Time in a multi-agent system is represented as a discrete set of time points and a relation over these points. This relation determines an order over the set of time points. The order forms a tree like structure. Time is viewed as having a single linear history and multiple possible futures. The possible futures represent the way the multi-agent system can possibly change. These possibilities include both the possible actions taken by the agents and the uncertainty in the world. Such uncertainty is with respect to the possible success or failure of an event.

We thus adopt a change based model of time and only consider time points that are relevant for the changing world. That is, we do not consider time points if no change has occurred. The order over the time points is backwards-linear.

**Why branching time?** The representation of possible futures is required for the purpose of modelling the decision making process of an agent and the chance of actions failing. This is further emphasized when the agent is faced with multiple goals to be achieved.

### 5.2.4 A Model of an Agent

Note that an agent is one part of the multi-agent system. The agent has only a limited knowledge about the world. Like in the single agent approach described above, the agent is modelled through its mental attitudes. These are introduced as special modalities that represent the beliefs, goals, and intentions of the agent. We thus model the mental attitudes as accessibility relations over the set of worlds. Given that the multi-agent system includes multiple agents the model includes such relations for each of the agents in the multi-agent system.
5.3 Modelling Air Missions with a Multi-Agent System

Similar to the single-agent approach, when using the multi-agent approach one has to specify what is the behavior required from each of the agents in the system. The primary difference is that one can now reason about the mental state of other agents. Furthermore, the maintenance of the mental state of one agent with respect to another agent’s mental state is facilitated by the underlying model.

Again, we start by using the analysis provided in Section 2.2.2. We then proceed through a process of refinement until we get a full specification of all the agents in the system. The analysis in this case will result in the same required behavior and identification of agents as in the single-agent case (see Chapter 4).

The way the relationships between agents are modelled and the way these relationships affects the plans is still identical to the single agent case. Here we only describe the modification to the design with respect to the mental state of the agents and the way coordination is implemented using these mental states.

5.3.1 Mental State

Here again each of the agents is independent. Nevertheless they have the ability to reason about the mental state of other agents (including their own). In particular they can identify actions performed by other agents and recognize the intention behind these actions.

Issues of consistency, in a single agent, between such mental states are now handled by the underlying model. For example, if one of the hornets believes that the FC believes that his partner is under threat (e.g., $\text{BEL}(\text{hornet1, BEL}(\text{FC, underthreat(hornet2))))$) he himself can adopt the belief that his partner is under threat. If for some other reason he believes that this is not the case then the system may identify inconsistencies between the two beliefs. The agent can react to these inconsistencies and try to resolve them in some way (e.g., inform the FC that his partner is fine).

Note that each agent still has to store information about its beliefs about the relationships with other agents. In particular the current command, control, and communication relationships. The way these beliefs affect its behavior will be “hard wired” into the the agent’s plans.

If the Attack can not operate relatively autonomously then the relationships will also include control relationship between the FC and the Attack. This control will only be with respect to the choice of tactics (i.e., plans) adopted by the attack. We say that the FC and Attack pair have a “close control” relationship. This relationship will have to be implicitly modelled as one of the beliefs of the agent.

Given that different relationships can hold for different goals, the agent has to store in its database the relationships and the goals they are applicable for. It is also important to note that in the presence of recognition of actions and mental states agents could now reason about more advanced interactions. For example they could consider assisting other
agents in achieving their goals, reasoning about the partial knowledge of other agents, and coordinating their intentions.

5.3.2 Plans and Coordination

The plans available to the the 2 hornets are identical to the plans described in the single agent case (see Chapter 4). The primary difference is that coordination can be based on an explicit representation of the mental attitudes of the two agents.

The designer is still required to ensure that both agents execute their respective plans at the same time. Nevertheless the agents the agents can interact with each other and coordinate their activities. One hornet can thus reason about the intentions of the other hornet and adjust its behavior accordingly. The agents can also communicate their mental attitudes directly allowing for the “synchronization” of the individual mental attitudes.

The designer of the system is still faced with the problem of joint decision making. Although there is no joint decision making, the designer can specify the agents in a way in which they dynamically coordinate their activities and make joint decisions. As an example, when the pair of hornets are required to assign the targets to each other they can communicate to each other which target they believe is closer to each one of them. They can then exchange information to adjust and synchronize their beliefs and then assign the targets based on an agreed belief.

If communication is not possible then the wing can observe the actions of the lead, reason about the intentions of the lead, and then adjust its behavior so the intentions will be compatible. The designer can specify such reasoning only because the underlying model allows for an explicit representation of the mental attitudes of different agents.

Note that although the agents can reason about the coordination, they are doing so from first principles. That is, they do not have plans for the pair as a single unit which specifies the coordinated activity to be adhered to by the underlying model. Rather, the coordination process is embedded in the existing plans and has to be reasoned about by the agents. Again the designer has to implicitly specify this coordination behavior.

Note again that the effect of the inter-agent relationship is “hard-wired” into the plans. It again follows that the designer would be required to specify multiple plans that achieve the same objectives under different relationships. Furthermore, to allow for flexible and dynamic inter-agent relationships the designer must specify in detail the mechanisms for initiating, identifying, and bringing about the change in these relationships.

Such specifications may include synchronization of the mental state of multiple agents. Given that the agents are able to identify the mental states of each other such synchronization is simpler.
5.4 The Formal Semantic Model of Multi-Agent Systems

We adopt a possible worlds branching-time tree structure in which there are multiple possible worlds and each world has a branching-time tree structure. Multiple possible worlds model the environment as viewed by the agents and are a result of the agents’ lack of complete knowledge about the environment. Within each world, the branching future represents the choice of actions available to the agents and the chance of these actions failing.

Again let us first define an interpretation of the language to be an extension of a standard Kripke interpretation of possible worlds. The extension involves each possible world being a temporal structure. The edges are sets of agent-event pairs representing the events performed by the relevant agents between these two time points.

**Definition 5**

Given a set $T$ of time points, a binary relation $\prec$ on $T$ which is total, transitive and backward-linear, a set of agent constants $A$, a set $E$ of events, and an edge function $C$ that maps successor-restricted time point pairs to sets of agent-event pairs, a *world* or *time tree*, $w$ is a tuple $(T_w, \prec_w, A, E, C_w)$, where $T_w \subseteq T$ is a set of time points in the world $w$ and $\prec_w$ and $C_w$ are the same as $\prec$ and $C$ respectively, restricted to time points in $T_w$.

Note that we associate with each time interval the set of agent-event pairs. We thus say that “the agent has performed the event between these time points”. Given a time interval $\langle t_i, t_j \rangle$ and an agent-event pair $\langle a, e \rangle$ that is associated with it, we refer to $t_i$ as the start time of the event and to $t_j$ as the end time of the event.

Also note that we restrict the edge function to consider only successor-restricted time point pairs. That is, we allow for events to be performed only over time point pairs in which the end time of the event is a successor of the start time of the event.

**Definition 6**

An *interpretation* of a multi-agent system, $M^{mas}$, is a tuple:

$$M^{mas} = \langle A, E, T, \prec, C, W, B_i, G_i, I_i, U, \Phi \rangle$$

$A = \{a_1, \ldots, a_n\}$ is a set of agent constants, $E$ is a set of events, $T$ is a set of time points. $\prec$ is a total, transitive, backwards-linear binary relation on $T$. $C$ is an edge function that maps successor-restricted time point pairs to sets of agent-event pairs. More formally, $C : \prec^* \to 2^{A \times E}$. Intuitively, for any two time points for which the edge function $C$ is defined, it represents the events performed by the relevant agents between those time points. $W$ is a set of worlds with respect to $T$, $\prec$, $E$, and $C$.

The accessibility relations, $B_i$, $G_i$, and $I_i$, where $i \in A$ map a current state of the agent (i.e., a time point in a world) to the belief-, goal-, and intention-accessible worlds of the agent $i$, respectively. More formally, $B_i \subseteq W \times T \times W$ and similarly for $G_i$ and $I_i$. 

© Copyright Gil Tidhar, 1999
U is the universe of discourse and $\Phi$ is a mapping of first-order syntactic entities to relations over U for any given world and time point.

An agent $a$ has a belief $\phi$, denoted by $\text{BEL}(a, \phi)$, at time point $t$ if and only if $\phi$ is true in all the belief-accessible worlds of the agent at time $t$ (and similarly for goals and intentions). Satisfaction of formulas is denoted by $\models$. In our model satisfaction is with respect to the structure $M^{\text{mas}}$, a world $w$, and a time point $t$. The expression $M^{\text{mas}}, w_t \models \phi$ should be read as “structure $M^{\text{mas}}$ in world $w$ and time point $t$ satisfies $\phi$”.

Definition 7

$M^{\text{mas}}, w_t \models \text{BEL}(a, \phi) \iff \forall w' \text{ such that } B_a(w, t, w'), M^{\text{mas}}, w'_t \models \phi$;

$M^{\text{mas}}, w_t \models \text{GOAL}(a, \phi) \iff \forall w' \text{ such that } G_a(w, t, w'), M^{\text{mas}}, w'_t \models \phi$; and

$M^{\text{mas}}, w_t \models \text{INTEND}(a, \phi) \iff \forall w' \text{ such that } I_a(w, t, w'), M^{\text{mas}}, w'_t \models \phi$.

As in the case of a single agent model one can impose semantic constraints between the different mental attitudes to define a variety of types of agent rationality (see the work of Rao and Georgeff [90] for further details).

### 5.5 Axioms

We extend the axioms described in Chapter 4 to include references to the relevant agents. That is, each of the mental attitudes, belief, goal, and intention, now include a reference to the agent that believes, has a goal towards, or intends to bring about, the particular formula considered.

These extensions are with respect to the standard K-axioms as they apply to the BDI model, i.e., ($B$-K), ($G$-K), and ($I$-K), as well as the BDI generalization rules, i.e., ($B$-Gen), ($G$-Gen), and ($I$-Gen). Similarly we modify the BDI-Consistency rules, ($BC0$), ($BC1$), ($GC0$), ($GC1$), ($IC0$), and ($IC1$) to include the relevant references to the agents.

For a given set of agents $\mathcal{A} = \{a_1, \ldots, a_m\}$ we replace the ($B$-K) axiom with a class of axioms ($B$-$K_{a_i}$) where $a_i \in \mathcal{A}$. We similarly replace the other axioms and rules mentioned above with the relevant classes of axioms and rules. This is similar to the class of $K_i$ axioms introduced by Halpern and Moses [54].

Following the nomenclature adopted in Chapter 4 the above axiom system is denoted $BGI^{\text{mas}}_{\text{CTL}}$.

### 5.6 Soundness and Completeness

The $BGI^{\text{mas}}_{\text{CTL}}$ axiom system and model described above are an extension of the single agent model described in the previous chapter. Here we will modify the proof of soundness and completeness for a single agent model to take into account the new concepts introduced.
Primarily the differences between the two models are: (a) the introduction of multiple accessibility relations for each of the agents; and (b) the association of agents with the particular events that occur. Note also that in the notation used here $R$, $R^q$, $L$, and $L^q$, are replaced by $\prec$, $\prec^q \Phi$, and $\Phi^q$, respectively.

We can proceed by defining $\text{BGI}_{\text{CTL}}^\text{mas}$-tableau, a quotient construction, and pseudo-$\text{BGI}_{\text{CTL}}^\text{mas}$-tableau where the accessibility relations for a single agent are replaced by the relevant relations for multiple agents. As defined in Section SAS-small-model-theorem, for a given formula $\phi_0$ the function $\text{ecl}(\phi_0)$ returns the extended closure of $\phi_0$.

Let us now prove the small model theorem for $\text{BGI}_{\text{CTL}}^\text{mas}$.

**Theorem 3** Let $\phi_0$ be a BGI $\text{CTL}$ formula of length $n$, then the following equivalences hold:

1. $\phi_0$ is $\text{BGI}_{\text{CTL}}^\text{mas}$-satisfiable.;
2. $\phi_0$ has a model $\mathcal{M}^\text{mas}$ with a finite branching in each world bounded by $O(n)$;
3. $\phi_0$ has a finite pseudo-$\text{BGI}_{\text{CTL}}^\text{mas}$-tableau of size $\leq \exp(n)$;
4. $\phi_0$ has a finite model $\mathcal{M}^\text{mas}$ of size $\leq \exp(n)$.

**Proof:** The proof of the CTL component is identical to that given in the proof given in Section 4.6 for (Theorem 1). We only need to make the appropriate changes to the proof that $(2) \rightarrow (3)$ and show that the models in $\mathcal{M}^\text{mas}$ satisfy the class of consistency rules $(\text{BC0}_a_i), (\text{BC1}_a_i), (\text{GC0}_a_i), (\text{GC1}_a_i), (\text{IC0}_a_i)$, and $(\text{IC1}_a_i)$.

Let $\mathcal{M}^\text{mas}$ be a class of structures with $M \in \mathcal{M}^\text{mas}$ such that $M, w_t \models \phi_0$. We show that $M^q = M/ \equiv \text{ecl}(\phi_0)$ is a pseudo-$\text{BGI}_{\text{CTL}}^\text{mas}$-tableau.

We show that the quotient construction satisfies $(\text{BC0}_a_i)$. The other conditions in the class $(\text{BC0}_a_i)$ can be proved in the same way thus proving that $\mathcal{M}^\text{mas}$ satisfy the class of consistency rules $(\text{BC0}_a_i)$. The other classes of conditions can be proved in the same way.

Let $\text{BEL}(a_1, \phi) \in \Phi(w_t)$ and $\mathcal{B}_{a_1}(w, t, v)$. From the quotient construction we have $M^q, [w_t] \models \text{BEL}(a_1, \phi)$ and $\mathcal{B}_{a_1}^q([w_t], [v_t])$. From the definition of belief we have $M^q, [v_t] \models \phi$ and as a result $\phi \in \Phi(v_t)$.

From the construction the size of the model can be shown to be $l \times N^2$ where $l$ is the number of eventualities and $N$ is the number of nodes in the model. The number of nodes $N \leq 2^n$ where $n$ is the length of $\phi_0$. We thus get that the size of the model $\leq \exp(n)$.

Note that this number is independent of the number of agents but rather depends on the complexity of the belief, goal, and intention accessibility relations.

Let us now describe the algorithm for constructing a pseudo-$\text{BGI}_{\text{CTL}}^\text{mas}$-tableau. Again the algorithm is essentially the same algorithm described in Section 4.7 with some changes made to Step 2(d) to take into account the multiple agents and the change in notation as follows:
5.6. SOUNDNESS AND COMPLETENESS

CHAPTR 5. MULTI-AGENT SYSTEMS

\[ \neg \text{BEL}(a, \neg \text{BEL}(b, p)) \]

\[ \neg \text{BEL}(b, p) \]

\[ \neg \text{p} \]

\[ \neg \text{B}_a \]

\[ \neg \text{B}_b \]

Figure 5.1: An example of a pseudo-BGI\textsubscript{CTL}\textsuperscript{mas} -tableau.

(d) Expanding elementary BDI formula: If node \( w_i \) is a leaf of the tree, \( \Phi(w_i) \) is not blatantly inconsistent, and \( \Phi(w_i) \) is fully expanded propositional tableau, then:

1. if \( \Phi(w_i) \) contains \( \neg \text{BEL}(a_{i_1}, \phi_1), \ldots, \neg \text{BEL}(a_{i_m}, \phi_m) \) then create \( m \text{ B}_{a_{i_j}} \)-successors of node \( w_i \), labelled with \( \text{BEL}^{-}(a_{i_j}, \Phi(w_i)) \cup \{\neg \phi_j\} \), where \( 1 \leq j \leq m \);

2. if \( \Phi(w_i) \) contains \( \neg \text{GOAL}(a_{i_1}, \phi_1), \ldots, \neg \text{GOAL}(a_{i_m}, \phi_m) \) then create \( m \text{ G}_{a_{i_j}} \)-successors of node \( w_i \), labelled with \( \text{GOAL}^{-}(a_{i_j}, \Phi(w_i)) \cup \{\neg \phi_j\} \), where \( 1 \leq j \leq m \);

3. if \( \Phi(w_i) \) contains \( \neg \text{INTEND}(a_{i_1}, \phi_1), \ldots, \neg \text{INTEND}(a_{i_m}, \phi_m) \) then create \( m \text{ I}_{a_{i_j}} \)-successors of node \( w_i \), labelled with \( \text{INTEND}^{-}(a_{i_j}, \Phi(w_i)) \cup \{\neg \phi_j\} \), where \( 1 \leq j \leq m \).

As an example consider the simple tableau illustrated in Figure 5.1. Note that the links between the nodes are labelled with (and belong to) different accessibility relations.

Given a formula \( \phi \) and a structure \( M^{\text{mas}} = \langle A, \mathcal{E}, T, \prec, \mathcal{C}, W, B_i, G_i, I_i, U, \Phi \rangle \), we define \( |W| \) to be the number of worlds in \( W \), \( |\text{BGI}^{\text{mas}}| \) to be the total number of pairs in \( B_i, G_i, \) and \( I_i \), and \( |\phi| \) to be the length of \( \phi \).

Given the above construction we get that: for a given formula \( \phi \) and structure \( M^{\text{mas}} = \langle A, \mathcal{E}, T, \prec, \mathcal{C}, W, B_i, G_i, I_i, U, \Phi \rangle \), there is an algorithm for checking if \( \phi \) is satisfied in \( M^{\text{mas}} \) that runs in time \( O((|W| + |\text{BGI}^{\text{mas}}|) \times |\phi|) \). The proof for this is identical to the proof of (Proposition 3.1) described by Halpern and Moses [54] (pp. 337).

**Theorem 4** The \( \text{BGI}^{\text{mas}}_{\text{CTL}} \)-system is a sound and complete axiomatization with respect to \( \mathcal{M}^{\text{mas}} \).

(c) Copyright Gil Tidhar, 1999 84
Proof:

The proof is essentially identical to the proof provided in Section 4.8 for (Theorem 2). The only modification is with respect to Case 3, i.e., node \( n \) is an internal node of the tree and is a fully expanded propositional CTL tableau. Here we do not have \( B, G, I \)-successors but rather \( B_{a_i}, G_{a_i}, I_{a_i} \)-successors. Given that we have also changed the pseudo-tableau construction algorithm in a similar way the proof which is based on this construction does not change.

\[ \square \]
Chapter 6

Team-Oriented Systems

The multi-agent approach described above suffers from a number of shortcomings. In particular it does not allow the designer interested in the behaviour of the system as a whole to specify its coordinated behavior at a high level. Rather it requires the designer to consider all the low level details. Furthermore, there is no explicit representation of the joint mental attitudes of a group of agents. They are simply a particular (hard-wired) combination of joint mental attitudes of the agents that are part of the group.

To overcome these shortcomings we extend the BGI model of a multi-agent system described above to a BGI model of a team-oriented system. The team-oriented model changes the focus from a single agent (or a collection of agents) to a team. It also extends the notions of belief, goals, and intentions to include joint mental attitudes. Such joint attitudes include mutual beliefs, joint goals, and joint intentions. We thus regard the mental attitudes of a team as basic attitudes (or first class entities).

A team may have a sub-team relationship with other teams. We refer to teams in such a relationship as its sub teams. A team always has a sub-team relationship with itself. For the purpose of comparison with other related work we will refer to a team with no sub-teams, other than itself, as an agent or a primitive team. Similarly, one can also refer to the mutual beliefs, joint goals, and joint intentions of an agent simply as beliefs, goals, and intentions respectively.

In both the single and multi-agent models the behavior of the agents was determined only by the mental attitudes of the individual agents. In the multi-agent model coordinated behavior or the behavior of a group of agents had to be defined in terms of the behavior of the individuals within the group.

In a system with multiple agents the primary interest is in the behavior of the whole system as a unit or team. The focus on a team and its joint mental attitudes allows the change of focus from the behavior of the individual to the behavior of the group. The relationship between the joint mental attitudes of the team and the joint mental attitudes of its sub-teams will depend on the particular model which is best suited for the application.
6.1 Possible Worlds Semantics for Team-Oriented Systems: Informal Discussion

In this section, we informally discuss a possible-worlds semantics for a team-oriented system and the mental attitudes of teams. Let us first consider the different elements that should be considered for a model of a team-oriented system. Many of these elements are identical to those described when considering multi-agent systems (see Chapter 5). Here we will identify the similarities and elaborate on the differences. We will proceed to discuss the model adopted here and some motivation for this selection. In the following section we provide the formal semantic model for a team-oriented system.

The main difference between a team-oriented system and a multi-agent system is that the focus moves from a single agent in the system to a team. The team has mutual beliefs, joint goals, and joint intentions. Each team has sub-teams that cooperate in order to achieve the team’s goals.

A team-oriented system is thus composed of a set of teams, a particular universe of discourse (i.e., the domain), and information about this universe. Each team is uniquely associated with a team constant (i.e., the team name). Some of the teams are related by a sub-team relationship. As mentioned above, a team with no sub-teams, other than itself, is referred to as a primitive team or an agent. The teams operate in the context of time and a world. Changes in the team-oriented system happen through the occurrence of events.

The concept of domain is identical to that described in the model of a multi-agent system. As to the model of events, although it is primarily similar to the model of events in multi-agent systems there is one major difference: a team can perform an action.

**Why allow teams to perform actions?** We allow teams to perform actions so that our model will also cater for the holistic approach to teams and multi-agent systems. We thus allow the designer of a team-oriented system to determine whether coordination of a particular activity should be explicitly modelled at the level of mental state activity or could be left at the level of execution of that activity with “pre-compiled” coordination (see Searle [97] and Tidhar [122] for further discussion on the holistic approach and its advantages). Note though that a particular model of a team-oriented system may restrict actions only to primitive teams (i.e., agents). This will result in a model in which all joint behavior is reduced to the behavior of agents.

The model of time in a team-oriented system is identical to the model of time in a multi-agent system. As mentioned above, the primary difference between the model of a team-oriented system and a multi-agent system is the introduction of a team and the sub-team relationship between teams. We will now proceed to discuss these new concepts.
6.1. POSSIBLE WORLDS SEMANTICS FOR TEAM-ORIENTED SYSTEMS: INFORMAL DISCUSSION

6.1.1 A Model of a Team

As mentioned in Chapter 3, our model of a team is inspired by the work of Tuomela on social groups. We will now provide a short account of Tuomela’s work (a more detailed account can be found in Chapter 3). This will be followed by an informal description and motivation for the model of the team adopted here.

Tuomela’s Social Groups

In his past work Tuomela has primarily focused on notions of a social group, group goals, group intentions, group actions, and mutual beliefs [132, 134–136]. As described in Section 3.2.5, Tuomela suggests that a social group is a set of agents that:

1. mutually believe that they are a social group;
2. have a mutually believed procedure that enables them to adopt joint intentions;
3. are all willing to use this procedure; and
4. mutually believe that the group is willing to use this procedure.

A mutual belief of a set of agents G in a proposition φ is defined to be that every member of G believes that φ and also that G mutually believes φ. This is a self-referential definition similar to that provided by us in previous work [69].

Note again that, Tuomela’s approach is based on the idea that the agent is the basic social unit. Other social entities are thus composed of basic social units that are associated by some set of interrelations. These inter-relationships are expressed as a procedure that links the mental attitudes of the members to form the joint mental attitudes of the group.

The Approach to Teams Adopted Here

In our approach to the definition of a team we start with the work of Tuomela and modify it appropriately. As mentioned above the work of Tuomela has primarily focused on groups of humans. This has forced him to consider problems such as lack of motivation or will, and the availability of a procedure that enables a group of humans to adopt joint intentions. In the work presented here we consider groups of artificial agents. Problems that arise due to lack of motivation or will can thus be ignored. This removes the need for part (3) of Tuomela’s definition.

In addition we are concerned with multi-agent systems that are centrally designed. We can thus assume that a procedure that enables a group of artificial agents to adopt joint mental attitudes will be part of the system. This removes the need for parts (2) and (4) of Tuomela’s definition.

As to the question of a holistic model of social entities, here we change the basic idea that underlies the Tuomela model. That is, we view the group (i.e., the team) as the basic social unit. Such a unit has basic interrelations with other units - the sub-team relation.
6.1. POSSIBLE WORLDS SEMANTICS FOR TEAM-ORIENTED SYSTEMS:
INFORMAL DISCUSSION

We thus adopt the interrelation aspects in Tuomela’s definition of a social group. Given the above assumptions we restrict them only to the mutual belief of the members (i.e., part (1) of Tuomela’s definition). Note that in the context of this approach an agent is a special case of a team - it is a team with no sub-teams.

A team is one part of the team-oriented system. The team has only a limited knowledge about the world. Like in the single agent approach described above, the team is modelled through its mental attitudes. These are introduced as special modalities that represent the beliefs, goals, and intentions of the team. We thus model the mental attitudes as accessibility relations over the set of worlds. Given that the team-oriented system includes multiple teams, the model includes such accessibility relations for each of the teams in the team-oriented system.

As mentioned above, some teams have a sub-team relationship with other teams. We view the sub-team relationship as mutual beliefs held by the respective teams about a particular state of the world, that is, their respective mutual belief that they are in a sub-team relationship. For the sub-team relation to hold between the teams, \( \tau \) and \( \tau' \), we require that both teams mutually believe that the proposition stating that they are in a sub-team relation is true. We say that a team \( \tau' \) is a sub-team of team \( \tau \) if and only if:

1. The team \( \tau \) mutually believes that the team \( \tau' \) is its sub-team; and
2. The team \( \tau' \) mutually believes that it is a sub-team of the team \( \tau \).

It is important to note here that two teams having a sub-team relationship is simply a state of mind for them. In our model this state of mind is based on the team’s mutual beliefs about the state of the world. Furthermore, note that the sub-team relationship does not represent an aggregation relationship. That is, a team is an atomic unit and is not “composed” of its sub-teams.

Unlike previous work [69] and Tuomela’s approach [135], the model described here does not necessarily define the mental attitudes of the team in terms of the mental-attitudes of the sub-teams. Rather, the team’s mental attitudes are primitive notions for the team. Nevertheless such relationships (e.g., an authority system) can be modelled using axioms that relate the two joint mental attitudes.

Team-oriented systems that enforce the mental attitudes of the team to be in some particular relationship to the mental attitudes of the sub-teams are modelled as a special case. A detailed discussion of such systems can be found in Chapter 7.

As a software component of a distributed computer system a team has its own physical realization. This includes its mutual beliefs, joint goals, and joint intentions. A discussion on the way teams are implemented can be found in Chapter 10.

**Why have teams as first class entities and not defined to be a group of agents?**

There are a number of reasons for having teams as first class entities. They can generally be categorized as either related to the expressiveness of the model or to the complexity of the corresponding logic.
Expressiveness: By not putting any restrictions on the sub-team relationship we allow a team to change its sub-teams dynamically without changing the “identity” of the team.\footnote{Also note that unless specified otherwise a team with some sub-teams can change to a team with no sub-teams and vice versa. Recall that a team with no sub-teams is like a single agent. The philosophical meaning of such a feature is unclear and not the main focus of this work. Obviously one can restrict the models to such models in which that feature is not permitted.} We thus separate the notion of a team from the notion of a set of agents or a set of sub-teams.\footnote{Obviously if some aspect of the team’s behavior or joint mental attitudes are dependent on the behavior or attitudes of the sub-teams then changing the sub-teams may affect the behavior or attitudes of the team.}

Note that this approach to the notion of a team differs from the approach adopted by Halpern and Moses\footnote{Copyright Gil Tidhar, 1999} [54] in their seminal work on knowledge in distributed systems. Here teams are first class entities represented explicitly in the model. Furthermore, we do not assume any particular relationship between the joint mental attitudes of a team and the joint mental attitudes of its sub-teams. The Halpern and Moses approach is a special case in our approach. See Chapter 7 for a detailed comparison between the two approaches.

Reasoning about team membership: Separating the notion of a team from its set of sub-teams allows us to provide mechanisms for reasoning about team membership without having to alter the properties of the team. In particular one can view the sub-teams as one part of the means the team has towards achieving its goals. The team can thus decide to add (or remove) a sub-team to facilitate the achievement of its goals.

Specification of joint mental attitudes: Note that our main objective in developing a system with multiple agents is so it could exhibit complex behavior in an uncertain environment. Within such a system the behavior of each sub-system will primarily depend on the required functionality and its particular environment. Here we refer to each of the sub-systems as a team. Having teams as first class entities allows us to attach to each team its own mental attitude. We can then consider different operational semantics to the joint mental attitudes of each team. The decision about the particular operational semantics will depend on the required behavior of the team and its operating environment. In addition this semantics may depend on the inter-team relationships, i.e., the relationships between its sub-teams (or sub-systems).

Modularity and Encapsulation: Having teams as first class entities allows the designer of a system to reason about a team as a unit. The designer need not consider the composition of the team until such information is required. The team is thus an encapsulation of the details of the operation of its sub-teams and of the coordinated
behavior of the sub-teams. The information about the composition of the team can be done at a later stage.

**Specification of coordinated behavior:** Recall that we use mental attitudes to specify the coordinated behavior of a team of agents. Defining teams as first class entities with their own mental attitudes allows us to specify a variety of types of coordinated behavior. This is done by specifying different relationships that exist between the joint mental attitudes of the teams and the joint mental attitudes of its sub-teams.

**Complexity:** When discussing the complexity of the model we consider two aspects: (1) the number of entities and relationships that are included in the model; and (2) the complexity of performing a particular task using the model. For the first aspect, note that in a system with multiple agents the number of teams that could be formed far exceeds the number of agents in the system. Nevertheless for any required complex group behavior, the change from an agent-oriented to a team-oriented model does not entail a change in the number of entities or relationships that should be included in the model. In particular one would be required to model both the mental attitudes of the team and the relationship between such attitudes and the attitudes of the agents that are part of the team.

For the second aspect, the tasks that we would consider would be: (1) developing a structure (or model) that satisfies a given formula; and (2) given a model and a formula checking if that formula is satisfied in that model. For both tasks the complexity would be considered in terms of the time required (represented as number of operations) and the space required (represented in terms of memory) as a function of the size of the given formula and structure (see the work of Hopcroft and Ullman [57] for further details on complexity).

When considering such complexities, we get that the upper bound on the complexity of performing these tasks in a team-oriented model is equivalent to the upper bound on the complexity of performing these tasks in an agent-oriented model. Note though that our main interest here is in modelling the joint mental attitudes of teams of agents. We thus consider the complexity of performing these tasks with respect to a particular mental attitude of a team.

In a team-oriented system we have teams as first class entities with ascribed joint mental attitudes. In the simplest team-oriented model the joint mental attitudes of the team are independent of the joint mental attitudes of the sub-teams. Checking a particular joint mental attitude of a team will thus be as complex as checking the mental attitude of a single agent in an agent-oriented system. This check will
only depend on the number of possible worlds considered for the particular mental attitudes.

Note that in an agent-oriented system the only first class entities are agents and hence a team is defined in terms of the agents that are members of the team. It follows that in such a model checking the mental attitudes of the team always entails checking the mental attitudes of all its members.

In the more complex cases we can specify that for some teams the joint mental attitudes of the team are related to the joint mental attitudes of some of its sub-teams. For such teams checking a particular joint mental attitude will entail checking that joint mental attitude in the team and also in the relevant sub-teams. In such a model the worst case scenario would be that we are required to check the joint mental attitudes of all the members of the team (similar to the agent-oriented case). Note that this is a worst case scenario but may not necessarily be the most common scenario in a team-oriented system.

Given the modular approach to the specification of the relationship between the joint mental attitudes of the team and the joint mental attitudes of the sub-teams we also obtain a similar “modularity” in the complexity of validating our model.

### 6.2 Syntax

The syntax of a team-oriented system described here extends the syntax described by us in the previous section and in previous work [69]. Again similar to CTL, there are two types of formulas in the logic: state formulas (these are evaluated at a given time point in a given world) and path formulas (these are evaluated along a given path in a given world). The main difference is that we now change the basic entities from being centred around the view of a single agent (even if in a multi-agent world) to a view of teams of agents.

We modify the definition of a state formula used in the multi-agent model. We do so by replacing the operators for belief, goal, and intention with the respective operators for mutual belief, joint goal, and joint intention. More formally:

- if $\phi$ is state formula and $\tau$ is a team then $\text{MBEL}(\tau, \phi)$, $\text{JGOAL}(\tau, \phi)$ and $\text{JINTEND}(\tau, \phi)$ are state formulas; and

- if $\tau$ and $\tau'$ are teams then $\text{subteam}(\tau, \tau')$ and $\text{SUBTEAM}(\tau, \tau')$ are state formulas;

The operators $\text{MBEL}$, $\text{JGOAL}$, and $\text{JINTEND}$ represent, respectively, the mutual beliefs, joint goals, and joint intentions of the relevant team.

Like any other proposition, the truth value of $\text{subteam}(\tau_i, \tau_j)$ in a particular state of the world is determined by a valuation function. It represents a particular aspect of the world in which two teams are in a sub-team relationship in that world.
6.3. MODELLING AIR MISSIONS WITH A TEAM-ORIENTED SYSTEM

Two teams are in a sub-team relationship if they both mutually believe that they are in such a relationship. The operator $\text{SUBTEAM}(\tau, \tau')$ represents a particular “state of mind” of the teams $\tau$ and $\tau'$. It involves the mutual beliefs of the teams $\tau$ and $\tau'$ with respect to the subteam proposition. It indicates the condition in which $\tau'$ is a sub-team of $\tau$. The truth value will depend on the semantics of mutual belief (see following section).

Note that the joint mental attitudes are first class entities in the language. Unlike our previous work [69] they are not defined in terms of other operators.

6.3 Modelling Air Missions with a Team-Oriented System

The change from a multi-agent system to a team oriented system allows (and requires) the designer to specify the teams that will take part in the mission, their coordinated behavior, and their sub-teams. The command, control, and communication relationships between teams and their impact on the team behavior would still have to be specified implicitly.

When using the team-oriented approach one has to specify what behavior is required from each of the teams in the system. The primary difference from the multi-agent approach is that one can now reason about the mental state of the team as a whole. Furthermore, one can specify the (coordinated) team behavior and let the underlying system handle the coordination.

We will first describe the teams that are part of this example and the sub-team relationships between them. We will then proceed to describe the relationships between the teams. We will conclude with a description of the team plans (i.e., team behavior) and the way the relationships between the team affect the team plans.

6.3.1 Teams

In the air mission modelling example there are two primary teams: Blue force and Orange force. We will focus here on the Blue force and its sub-teams.

At the initial stage the Blue force has a sub-team relationship with a Sector Air Defence, AEW&C aircraft, AP-3C aircraft, and ADGE. The Sector Air Defence has a sub-team relationship with a SADC, and 2 pairs of hornets. We refer to these pairs as an Attack and a CAP respectively. The SADC is an agent and each hornet in the pair is also an agent. The AEW&C has a sub-team relationship with an MC and two FCs. In this scenario we view the AP-3C and ADGE as agents.

It is important to note that as described above the teams change over time. In particular the two FCs on-board the AEW&C each join the Attack and CAP. The two teams are joint by the MC to form the AEW&C team. All in all as part of the Blue team we have the following teams:

- AEW&C team;
• Attack team;
• CAP team;
• a single SADC agent;
• four hornet agents;
• a single MC agent;
• two FC agents;
• a single AP-3C agent; and
• a single ADGE agent.

6.3.2 Joint Mental Attitudes

Unlike the multi-agent case, here each of the teams is a first class entity. We need to describe the relationships between the joint mental attitudes of the team and the joint mental attitudes of the sub-teams. We adopt the following model: (1) each team will have knowledge about its relationship with other teams and in particular each team will have a commanding team; (2) if the commanding team has a particular joint mental attitude then so will the team which it commands; (3) if a team has a joint goal or joint intention then so will each sub-team have those joint mental attitudes.

Note that the relationship between the joint mental attitudes of different teams will depend on their beliefs as to the social relationships between them. Issues of consistency between the mental state of the commanding team and commanded team must be handled by the designer.

For example, the MC commands the Attack. The MC may believe that there is a target based on sensory input from the radar. It may adopt a joint goal to intercept this target. When considering the Attack team, the target may be out of radar range. This Attack has no mutual beliefs about the existence of the target. It can not adopt a joint goal to intercept the target until the mutual beliefs of the MC and the Attack are made consistent.

Note that each team still has to store information about its mutual beliefs about the relationships with other teams. In particular the current command, control, and communication relationships. These relationships identify who determines the goals of the team, how are decisions made in the team, and what information is exchanged. The way these beliefs affect the team’s behavior will be “hard wired” into the team’s plans. Again such relationships may change over time.

Note that the use of joint mental attitudes allows the developer to overcome some of the problems of coordinating the mental attitudes of multiple sub-teams. Nevertheless this does not remove the need to coordinate the way the sub-teams make joint decisions or control their joint activities. These processes have to be explicitly specified by the developer. If we require the teams to dynamically alter these processes then the developer must also specify in detail how will these changes take place.
6.3.3 Team Plans and Coordination

Unlike the multi-agent case, the designer can now specify the behavior of the coordinated components at the top level. That is, we start by analyzing the air mission scenario to identify the required behavior of the system as a whole. We then specify this behavior in the form of team plans. The way the relationships between teams are modeled and the way these relationships affects the plans is still similar to the multi-agent case.

Given the team plan, each of the sub-teams involved in the execution of the sub-goals is only required to know how to achieve these goals. It does not have to have knowledge of the team plan. The underlying system will ensure that the coordinated activity of the team as a whole is achieved.

Team decision making is all done at the level of the team without the need to involve the sub-teams. For example, the Attack can choose to adopt a pincer intercept in which one hornet approaches that target from the left and the other hornet from the right. The execution of this intercept and the coordination of the two hornets is done at the level of the Attack team. Note though that the designer still has to ensure that the joint intention of the commanding team and the joint intention of the commanded team are consistent.

6.4 The Formal Semantic Model of Team-Oriented Systems

As mentioned above, the model adopted here is very similar to the model of a multi-agent system. The main difference is with respect to: (1) the focus on teams; and (2) the introduction of a sub-team relationship and its impact on the various mental attitudes.

Again we adopt a \textit{possible worlds branching-time tree structure} in which there are multiple possible worlds and each world has a branching-time tree structure. Multiple possible worlds model the teams’ lack of complete knowledge about the environment. Within each world, the branching future represents the choice of actions available to the team and the chance of success or failure of these actions.

6.4.1 The Model

Let us first define an interpretation of the language to be an extension of a standard Kripke interpretation of possible worlds. The extension involves each possible world being a temporal structure. The edges are sets of team-event pairs representing the events performed by the relevant teams between these two time points.

\textbf{Definition 8}

Given a set \( T \) of time points, a binary relation \( \prec \) on \( T \) which is total, transitive and backward-linear, a set of team constants \( T \mathcal{M} \), a set \( \mathcal{E} \) of events, and an edge function \( \mathcal{C} \) that maps successor-restricted time point pairs to sets of team-event pairs, a \textit{world} or \textit{time}
tree, \( w \) is a tuple \( \langle T_w, \prec_w, T_M, E, C_w \rangle \), where \( T_w \subseteq T \) is a set of time points in the world \( w \) and \( \prec_w \) and \( C_w \) are the same as \( \prec \) and \( C \) respectively, restricted to time points in \( T_w \).}

The definition of successor time points is identical to that given in Section 5.4. Again we restrict the edge function to allow for events to be performed only over successor time points.

**Definition 9**

An interpretation of a team-oriented system, \( M^{tos} \), is a tuple:

\[
M^{tos} = (T_M, E, T, \prec, C, W, MB_i, JG_i, JI_i, U, \Phi)
\]

\( T_M = \{\tau_1, \ldots, \tau_n\} \) is a set of team constants, \( E \) is a set of events, \( T \) is a set of time points. \( \prec \) is a total, transitive, backwards-linear binary relation on \( T \). \( C \) is an edge function that maps successor-restricted time point pairs to sets of team-event pairs. More formally, \( C : \prec^s \rightarrow 2^{T_M \times E} \). Intuitively, for any two time points for which the edge function \( C \) is defined, it represents the events performed by the relevant teams between those time points. \( W \) is a set of worlds with respect to \( T \), \( \prec \), \( E \), and \( C \).

The accessibility relations, \( MB_i \), \( JG_i \), and \( JI_i \) where \( i \in T_M \) map a current state of the team (i.e., a time point in a world) to the mutual-belief-, joint-goal-, and joint-intention-accessible worlds of the team \( i \), respectively. More formally, \( MB_i \subseteq W \times T \times W \) and similarly for \( JG_i \) and \( JI_i \).

\( U \) is the universe of discourse and \( \Phi \) is a mapping of first-order syntactic entities to relations over \( U \) for any given world and time point.

Note again that we will at times refer to the above relations as function to the set \( \{True, False\} \). A team \( \tau \) has a mutual-belief \( \phi \), denoted by \( MBEL(\tau, \phi) \), in a world \( w \) at time point \( t \) if and only if \( \phi \) is true in all the mutual-belief-accessible worlds of the team at time \( t \) (and similarly for goals and intentions). Satisfaction of formulas is similar to satisfaction in the multi-agent model.

**Definition 10**

\[
M^{tos}, w_t \models MBEL(\tau, \phi) \text{ iff } \forall w' \text{ such that } MB_T(w, t, w'), M^{tos}, w'_t \models \phi;
\]

\[
M^{tos}, w_t \models JGOAL(\tau, \phi) \text{ iff } \forall w' \text{ such that } JG_T(w, t, w'), M^{tos}, w'_t \models \phi; \text{ and}
\]

\[
M^{tos}, w_t \models JINTEND(\tau, \phi) \text{ iff } \forall w' \text{ such that } JI_T(w, t, w'), M^{tos}, w'_t \models \phi.
\]

The sub-team relation between teams \( \tau \) and \( \tau' \) is based on the mutual beliefs held by both teams. We thus define the set \( ST \) to represent the sub-team relation. That is, it defines for a given world and time point the teams that are in a sub-team relationship. The format of the relation is \( ST \subseteq T_M \times T_M \times W \times T \).
We say that a team $\tau$ has a sub-team $\tau'$, in a world $w$ at time point $t$ if and only if the formula $\text{subteam} (\tau, \tau')$ is mutually believed by both teams. More formally we define the $ST$ relation as follows:

**Definition 11**

$$\forall w \forall t \ (\tau, \tau', w, t) \in ST \text{ if and only if }$$

$$\forall v : (w, t, v) \in MB_\tau \text{ we have } M^{tos}, v_t \models \text{subteam}(\tau, \tau') \text{ and }$$

$$\forall v : (w, t, v) \in MB_{\tau'} \text{ we have } M^{tos}, v_t \models \text{subteam}(\tau, \tau').$$

Note that $M^{tos}, w_t \models \text{MBEL}(\tau, \text{subteam}(\tau, \tau'))$ if $\forall w'$ such that $MB_\tau(w, t, w'), M^{tos}, w'_t \models \text{subteam}(\tau, \tau')$. Based on the above definition we define the $\text{SUBTEAM}$ operator to represent the sub-team relation. That is, it defines the pairs of teams that are in a sub-team relationship and is defined as follows:

**Definition 12**

$$M^{tos}, w_t \models \text{SUBTEAM}(\tau, \tau') \text{ if and only if } (\tau, \tau', w, t) \in ST.$$  

Note that there is no fundamental relationship between the joint mental attitudes of a team and the joint mental attitudes of the sub-teams (except a mutual belief about the subteam proposition). One can impose semantic constraints between the different joint mental attitudes of a team and the joint mental attitudes of the sub-teams. Examples of such constraints are described in Chapter 7.

Recall that the possible worlds reflect the team’s lack of complete knowledge about the environment. This lack of knowledge may include uncertainty about the current sub-teams. A model which includes both multiple possible sub-teams and semantic constraints between the different joint mental attitudes of a team and the joint mental attitudes of the possible sub-teams is very complex. To simplify the model we restrict our discussion to models in which the sub-team relation is consistent across multiple worlds but can change over time. More formally:

**Definition 13**

If $M^{tos}, w_t \models \text{SUBTEAM}(\tau, \tau')$ then $\forall v M^{tos}, v_t \models \text{SUBTEAM}(\tau, \tau').$

Note that the sub-team relation can change between two different time points. We also restrict our discussion to a sub-team relation with no cycles, i.e., we eliminate the possibility of a team being a sub-team of one of its own sub-teams. This restriction together with the finite number of teams ensures that given a particular team the number of teams that are associated with it via a sequence of sub-team relationships is finite.

Note that this definition of $ST$ is based on the respective $MB$ relations. In the remainder of this chapter we will refer to the $ST$ relation instead of the combination of mutual beliefs as defined above.
As in the case of a multi-agent model one can also impose semantic constraints between the different joint mental attitudes of each team and define variety of team rationalities. The discussion on team rationality is beyond the scope of this work. See the work of Rao and Georgeff [90] for further details on single agent rationality.

6.4.2 Syntactic and Semantic Conditions

We modify the axioms described in Chapter 5 to become joint mental attitudes and to include references to the relevant teams. That is, each of the mental attitudes, belief, goal, and intention, is replaced by a joint mental attitude, mutual belief, joint goal, and joint intention, respectively. Each of the joint mental attitudes includes a reference to the team that mutually believes, has a joint goal towards, or jointly intends to bring about, the particular formula considered. We also add relevant axioms for the sub-team relationship.

The extensions to the mental attitudes are with respect to the class of standard K-axioms as they apply to the model GI\textsubscript{K}\text{CTL}, i.e., \( (B-K_{a_i}) \), \( (G-K_{a_i}) \), and \( (I-K_{a_i}) \), as well as the class of GI generalization rules, i.e., \( (B-Gen_{a_i}) \), \( (G-Gen_{a_i}) \), and \( (I-Gen_{a_i}) \) where \( a_i \in A = \{a_1, \ldots, a_m\} \). Similarly we modify the class of GI-Consistency rules, \( (BC0_{a_i}) \), \( (BC1_{a_i}) \), \( (GC0_{a_i}) \), \( (GC1_{a_i}) \), \( (IC0_{a_i}) \), and \( (IC1_{a_i}) \) to include the relevant references to teams.

For a given set of teams \( T, M = \{\tau_1, \ldots, \tau_m\} \) we replace the class of axioms \( (B-K_{a_i}) \) with a class of axioms \( (MB-K_{\tau_i}) \) where \( \tau_i \in T, M \). We similarly replace the other classes of axioms and rules mentioned above with the relevant classes of axioms and rules, e.g., \( (JG-K_{\tau_i}) \), \( (JI-K_{\tau_i}) \), etc. Following the nomenclature adopted in Chapter 5 the above axiom system is denoted GI\textsubscript{K}\text{CTL}.

We now add the consistency rules for the sub-team relationship that exists between teams.

Sub-team Consistency Rules:

\[
\text{(STC0)} \quad \text{If } \text{SUBTEAM}(\tau_i, \tau_j) \in \Phi(w_i) \text{ and } (w, t, v) \in MB_{\tau_i} \cup MB_{\tau_j} \text{ then } subteam(\tau_i, \tau_j) \in \Phi(v_i)
\]

\[
\text{(STC1)} \quad \text{If } \neg\text{SUBTEAM}(\tau_i, \tau_j) \in \Phi(w_i) \text{ then } \exists v \text{ such that } (w, t, v) \in MB_{\tau_i} \cup MB_{\tau_j} \text{ and } \neg\text{subteam}(\tau_i, \tau_j) \in \Phi(v_i)
\]

In the first part of this section we provided a formal model for a team-oriented system. This model has focused on teams and introduced the sub-team relationship between teams. It has not yet provided much insight into the way such a model could be used to specify the behavior of a system with multiple agents.

As mentioned above the model can be used to specify the coordinated behavior of a group of agents. This coordination of groups of agents is based on the sharing of beliefs,
goals, and intentions. The model described above is the basis for the specification of a variety of system models that include different approaches to coordinated behavior. Such specifications are based on characterizing the relationships between the joint mental attitudes of the team and the joint mental attitudes of its sub-teams. In Chapter 7 we describe a number of axiom systems that specify different system models. We will now proceed to show the soundness and completeness of the above model.

6.5 Soundness and Completeness

The $\text{BGI}_{\text{CTL}}$ axiom system and model described above are an extension to the multi-agent model described in Chapter 5. Here we will modify the proof of soundness and completeness for a multi-agent model to take into account the new concepts introduced.

Primarily the differences between the two models are: (a) the change from agents to teams; and (b) the introduction of the sub-team relationship between teams.

We can proceed by defining $\text{BGI}_{\text{CTL}}$-tableau, a quotient construction, and pseudo-$\text{BGI}_{\text{CTL}}$-tableau where the accessibility relations for multiple agents are replaced by the relevant relations for multiple teams. Furthermore we add the relevant sub-team relations. We begin by defining a $\text{BGI}_{\text{CTL}}$-tableau:

**Definition 14**

Given a formula $\phi_0$, a $\text{BGI}_{\text{CTL}}$-tableau for $\phi_0$ is a tuple

$$\langle \mathcal{T}, \mathcal{E}, \mathcal{T}, \prec, \mathcal{C}, W, \mathcal{MB}_i, \mathcal{JG}_i, \mathcal{JI}_i, U, \Phi \rangle$$

(with $\phi_0 \in \Phi(w_t)$ for some $w \in W$ and some $t \in T_w \in T$) which meets the following conditions:

1. the propositional consistency rules (PC0-2);
2. the local consistency rules (LC0-2);
3. each eventuality is fulfilled;

Figure 6.1: An example of a pseudo-$\text{BGI}_{\text{CTL}}$-tableau.
4. the BGI\textsuperscript{tos}-consistency rules \((\text{MBC0}_{\tau_i})\), \((\text{MBC1}_{\tau_i})\), \((\text{JGC0}_{\tau_i})\), \((\text{JGC1}_{\tau_i})\), \((\text{JIC0}_{\tau_i})\), and \((\text{JIC1}_{\tau_i})\).

5. the sub-team consistency rules, \((\text{STC0})\) and \((\text{STC1})\).

We now define the following quotient construction for \(M\text{\textsuperscript{tos}}\):

**Definition 15**

The quotient construction for \(M\text{\textsuperscript{tos}}\) is similar to the quotient construction for \(M\text{\textsuperscript{mas}}\) with \(B\text{\textsuperscript{q}}a_i\), \(G\text{\textsuperscript{q}}a_i\), and \(I\text{\textsuperscript{q}}a_i\), replaced by \(MB\text{\textsuperscript{q}}\tau_i\), \(JG\text{\textsuperscript{q}}\tau_i\), and \(JI\text{\textsuperscript{q}}\tau_i\), respectively. We also add the following definition: \(\text{SUBTEAM}^-(\tau_i, X) = \{\text{subteam}(\tau_{j_1}, \tau_{j_2}) : \text{SUBTEAM}(\tau_{j_1}, \tau_{j_2}) \in X \text{ and } \tau_i = \tau_{j_1} \text{ or } \tau_i = \tau_{j_2} \}\).

A pseudo-BGI\textsuperscript{tos}-\text{CTL}-tableau is a pseudo-BGI\textsuperscript{mas}-\text{CTL}-tableau modified to include the relevant BGI\textsuperscript{tos}-consistency rules and the sub-team consistency rules. Let us now prove the small model theorem for BGI\textsuperscript{tos}\text{CTL}.

**Theorem 5** Let \(\phi_0\) be a BGI\text{\textsuperscript{CTL}} formula of length \(n\), then the following equivalences hold:

1. \(\phi_0\) is BGI\textsuperscript{tos}\text{CTL}-satisfiable.;
2. \(\phi_0\) has a model \(M\text{\textsuperscript{tos}}\) with a finite branching in each world bounded by \(O(n)\);
3. \(\phi_0\) has a finite pseudo-BGI\textsuperscript{tos}\text{CTL}-tableau of size \(\leq \exp(n)\);
4. \(\phi_0\) has a finite model \(M\text{\textsuperscript{tos}}\) of size \(\leq \exp(n)\).

**Proof:** The proof of the CTL component is identical to that given in the proof provided in Chapter 5 for **Theorem 3**. For the BGI component we need to replace the mental attitudes by joint mental attitudes and agents with teams and again the proof will be identical to that given in the proof given for Theorem 3.

Here we only need to make the appropriate changes to the proof that \((2)\rightarrow(3)\) and show that the models in \(M\text{\textsuperscript{tos}}\) also satisfy the sub-team consistency rules \((\text{STC0})\) and \((\text{STC1})\).

Let \(\text{SUBTEAM}(\tau_i, \tau_j) \in \Phi(w_i)\) and \((w,t,v) \in MB_{\tau_i} \cup MB_{\tau_j}\). From the definition of \(\text{SUBTEAM}\) and the quotient construction we have \(M', \ [w_i] \models \text{MBEL}(\tau_i, \text{subteam}(\tau_i, \tau_j))\) and \(\text{MBEL}(\tau_j, \text{subteam}(\tau_i, \tau_j))\). From the assumption \((w,t,v) \in MB_{\tau_i}\) or \((w,t,v) \in MB_{\tau_j}\). From the definition of \(\text{MBEL}\) we have \(M', [w_i] \models \text{subteam}(\tau_i, \tau_j)\). As a result \(\text{subteam}(\tau_i, \tau_j) \in \Phi(v_i)\).

From the construction, the size of the model can be shown to be \(l \ast N^2\) where \(l\) is the number of eventualities and \(N\) is the number of nodes in the model. The number of nodes \(N \leq 2^n\) where \(n\) is the length of \(\phi_0\). We thus get that the size of the model \(\leq \exp(n)\).
Note that this number is independent of the number of teams but rather depends on
the complexity of the mutual belief, joint goal, and joint intention accessibility relations.

Let us now describe the algorithm for constructing a pseudo-\(BGI_{\text{CTL}}\)-tableau. Again
the algorithm is essentially the same algorithm described in Chapter 5 with some changes
made to take into account multiple teams, the change in notation, and the introduction of
the sub-team relation. In particular we ensure that it always holds that a team is a sub-
team of itself. This is done by ensuring that a node which contains a formula that states
otherwise will be blatantly inconsistent. The changes are made to Step 2(d) as follows:

2(d) Expanding elementary BGI formula: If node \(w_t\) is a leaf of the tree, \(\Phi(w_t)\) is not
blatantly inconsistent, and \(\Phi(w_t)\) is fully expanded propositional tableau, then:

1. if \(\Phi(w_t)\) contains \(-\text{MBEL}(\tau_{i_1}, \phi_1) \ldots -\text{MBEL}(\tau_{i_m}, \phi_m)\) then create \(m \cdot \text{MB}_{\tau_{i_j}}\)-successors
   of node \(w_t\), labelled with \(\text{MBEL}^{-}(\tau_{i_j}, \Phi(w_t)) \cup \text{SUBTEAM}^{-}(\tau_{i_j}, \Phi(w_t)) \cup \{¬\phi_j\}\),
   where \(1 \leq j \leq m\);

2. if \(\Phi(w_t)\) contains \(-\text{JGOAL}(\tau_{i_1}, \phi_1) \ldots -\text{JGOAL}(\tau_{i_m}, \phi_m)\) then create \(m \cdot \text{JG}_{\tau_{i_j}}\)-successors
   of node \(w_t\), labelled with \(\text{JGOAL}^{-}(\tau_{i_j}, \Phi(w_t)) \cup \{¬\phi_j\}\), where \(1 \leq j \leq m\);

3. if \(\Phi(w_t)\) contains \(-\text{JINTEND}(\tau_{i_1}, \phi_1) \ldots -\text{JINTEND}(\tau_{i_m}, \phi_m)\) then create \(m \cdot \text{JI}_{\tau_{i_j}}\)-successors
   of node \(w_t\), labelled with \(\text{JINTEND}^{-}(\tau_{i_j}, \Phi(w_t)) \cup \{¬\phi_j\}\), where \(1 \leq j \leq m\);

4. if \(\Phi(w_t)\) contains \(-\text{SUBTEAM}(\tau_{i_1}, \tau_{i_j}) \ldots -\text{SUBTEAM}(\tau_{i_m}, \tau_{i_m})\) then create \(m \cdot \text{MB}_{\tau_{i_j}}\)-successors
   of node \(w_t\), labelled with \(\text{MBEL}^{-}(\tau_{i_j}, \Phi(w_t)) \cup \text{MBEL}^{-}(\tau_{i_j}, \Phi(w_t)) \cup \{¬\text{subteam}(\tau_{i_j}, \tau_{i_j})\}\),
   where \(1 \leq j \leq m\) and also create \(m \cdot \text{MB}_{\tau_{i_j}}\)-successors of node
   \(w_t\), labelled with \(\text{SUBTEAM}^{-}(\tau_{i_j}', \Phi(w_t)) \cup \text{MBEL}^{-}(\tau_{i_j}', \Phi(w_t)) \cup \{¬\text{subteam}(\tau_{i_j}, \tau_{i_j})\}\),
   where \(1 \leq j \leq m\);

Consider the simple tableau illustrated in Figure 6.1. Note that in this case the mutual
belief held by team \(a\) has no impact on node M1. Nevertheless the existence of the formula
\(\text{SUBTEAM}(a, b)\) forces the addition of the proposition \(\text{subteam}(a, b)\) to node M1. Now
consider the tableau illustrated in Figure 6.2. We have replaced the proposition \(p\) with the
proposition \(\text{subteam}(a, b)\). We get in node M0 the formula that \(b\) does not mutually believe
the proposition \(\text{subteam}(a, b)\). Given that it also holds that \(\text{SUBTEAM}(a, b)\) we get that
node M1 is blatantly inconsistent.

Given a formula \(\phi\) and a structure \(M^{\text{tos}} = \langle TM, \mathcal{E}, T, \prec, C, W, ST, \text{MB}_t, \text{JG}_t, \text{JI}_t, U, \Phi \rangle\),
we define \(|W|\) to be the number of worlds in \(W\), \(|BGI^{\text{tos}}|\) to be the total number of pairs
in \(\text{MB}_{\tau_{i_j}}, \text{JG}_{\tau_{i_j}}, \text{JI}_{\tau_{i_j}}\), \(|ST|\) to be number of sub-team relations, and \(|\phi|\) to be the
length of \(\phi\).

There is no principle difference in complexity computation between the algorithm
described above and the algorithm presented in Chapter 5. Nevertheless the number of
accessibility relations will be larger and hence the final complexity of checking the satisfi-
ability of a given formula may be larger.
Given the above construction we get that: for a given formula $\phi$ and structure $M^{\text{tos}} = (TM, \mathcal{E}, T, \prec, C, W, MB_i, \mathcal{JG}_i, \mathcal{JI}_i, U, \Phi)$ (where $ST$ is defined as above), there is an algorithm for checking if $\phi$ is satisfied in $M^{\text{tos}}$ that runs in time $O((|W| + |BGIT^{\text{tos}}| + |ST|) \cdot |\phi|)$.

To show that $BGICTL$ system is sound and complete we need to show that a formula $\phi_0$ is $BGICTL$-provable if and only if $\phi_0$ is marked ‘satisfiable’ in a pseudo-$BGICTL$-tableau. To show this we will need to prove the following two lemmas that state that in the pseudo-$BGICTL$-tableau if a label of node is inconsistent then the label of its $\prec$ and $MB_i$ predecessors are also inconsistent.

These two lemmas are similar to Lemma 1 and Lemma 2 presented by Rao and Georgeff [90] (pp. 41). The proof of the first lemma is identical to the proof given by Rao and Georgeff [90] and will not be provided here.

The proof of the second lemma in the case of joint goals and joint intentions is identical to the proof of the second lemma given by Rao and Georgeff. The proof of the second lemma in the case of mutual belief needs to be adjusted to take into account the introduction of the sub-team relationship and $\text{SUBTEAM}$ operator. The proof is provided below.

**Lemma 3** Let $\phi_m$ and $\phi_n$ be conjunctions of propositions in $m$ and $n$ respectively and $(n, m) \in \prec$ as constructed in the pseudo-$BGICTL$-tableau then if $\phi_m$ is inconsistent then $\phi_n$ is also inconsistent.

**Lemma 4** Let $\phi_m$ and $\phi_n$ be conjunctions of propositions in $m$ and $n$ respectively and $(n, m) \in MB_i$ as constructed in the pseudo-$BGICTL$-tableau then if $\phi_m$ is inconsistent then $\phi_n$ is also inconsistent.

**Proof: (Lemma 4)**

Suppose $(n, m) \in MB_i$. We have two options which arise from the construction:
1. For some formula $\neg \text{MBEL}^{c}(\tau_{i}, \psi_{k})$ in node $n$ such that $\text{MBEL}^{-}(\tau_{i}, n) = \{\phi_{1}, \ldots, \phi_{z}\}$ and $\text{SUBTEAM}^{-}(\tau_{i}, n) = \{\text{subteam}(\tau_{i}, \tau'_{1}), \ldots, \text{subteam}(\tau_{o}, \tau'_{v})\}$ we have, $\phi_{1}, \ldots, \phi_{z}, \text{subteam}(\tau_{l}, \tau'_{1}), \ldots, \text{subteam}(\tau_{l}, \tau'_{v})$, and $\neg \psi_{k}$ in node $m$.

2. For some formula $\neg \text{SUBTEAM}(\tau_{i}, \tau'_{j})$ in node $n$ such that $\text{MBEL}^{-}(\tau_{i}, n) = \{\phi_{1}, \ldots, \phi_{z}\}$ and $\text{SUBTEAM}^{-}(\tau_{i}, n) = \{\text{subteam}(\tau_{i}, \tau'_{1}), \ldots, \text{subteam}(\tau_{o}, \tau'_{v})\}$ or $\text{MBEL}^{-}(\tau'_{j}, n) = \{\phi_{1}, \ldots, \phi_{z}\}$ and $\text{SUBTEAM}^{-}(\tau'_{j}, n) = \{\text{subteam}(\tau_{l}, \tau'_{1}), \ldots, \text{subteam}(\tau_{l}, \tau'_{v})\}$ we have, $\phi_{1}, \ldots, \phi_{z}, \text{subteam}(\tau_{l}, \tau'_{1}), \ldots, \text{subteam}(\tau_{l}, \tau'_{v})$, and $\neg \psi_{k}$ in node $m$.

For both cases we have the following:

1. $\vdash \phi_{1} \rightarrow (\phi_{2} \rightarrow (\ldots (\phi_{z} \rightarrow (\text{subteam}(\tau_{i}, \tau'_{1}) \rightarrow (\text{subteam}(\tau_{i}, \tau'_{2}) \rightarrow (\ldots (\text{subteam}(\tau_{o}, \tau'_{v}) \rightarrow \psi_{k}) \ldots))) \{\text{Assumption that } \Phi_{m} \text{ is inconsistent and propositional reasoning}\}$

2. $\vdash \text{MBEL}(\tau_{i}, \phi_{1}) \rightarrow (\phi_{2} \rightarrow (\ldots (\phi_{z} \rightarrow (\text{subteam}(\tau_{i}, \tau'_{1}) \rightarrow (\text{subteam}(\tau_{i}, \tau'_{2}) \rightarrow (\ldots (\text{subteam}(\tau_{o}, \tau'_{v}) \rightarrow \psi_{k}) \ldots)))) \{\text{Generalization Rule MB-Gen}\}$

3. $\vdash \text{MBEL}(\tau_{i}, \phi_{1}) \rightarrow (\phi_{2} \rightarrow (\ldots (\phi_{z} \rightarrow (\text{subteam}(\tau_{i}, \tau'_{1}) \rightarrow (\text{subteam}(\tau_{i}, \tau'_{2}) \rightarrow (\ldots (\text{subteam}(\tau_{o}, \tau'_{v}) \rightarrow \psi_{k}) \ldots)))) \rightarrow (\text{MBEL}(\tau_{i}, \phi_{1}) \rightarrow (\text{MBEL}(\tau_{i}, \phi_{2}) \rightarrow (\ldots (\text{MBEL}(\tau_{i}, \phi_{z}) \rightarrow (\text{MBEL}(\tau_{i}, \text{subteam}(\tau_{i}, \tau'_{1})) \rightarrow (\text{MBEL}(\tau_{i}, \text{subteam}(\tau_{i}, \tau'_{2})) \rightarrow (\ldots (\text{MBEL}(\tau_{i}, \text{subteam}(\tau_{o}, \tau'_{v})) \rightarrow (\text{MBEL}(\tau_{i}, \psi_{k})) \ldots))) \{\text{From Axiom MB-K}\}$

4. $\vdash \text{MBEL}(\tau_{i}, \phi_{1}) \rightarrow (\text{MBEL}(\tau_{i}, \phi_{2}) \rightarrow (\ldots (\text{MBEL}(\tau_{i}, \phi_{z}) \rightarrow (\text{MBEL}(\tau_{i}, \text{subteam}(\tau_{i}, \tau'_{1})))) \rightarrow (\text{MBEL}(\tau_{i}, \text{subteam}(\tau_{i}, \tau'_{2})) \rightarrow (\ldots (\text{MBEL}(\tau_{i}, \text{subteam}(\tau_{o}, \tau'_{v})) \rightarrow (\text{MBEL}(\tau_{i}, \psi_{k})) \ldots))) \{\text{Propositional Reasoning}\}$

5. $\vdash \neg (\text{MBEL}(\tau_{i}, \phi_{1}) \land \text{MBEL}(\tau_{i}, \phi_{2}) \land \ldots \text{MBEL}(\tau_{i}, \phi_{z}) \land \text{MBEL}(\tau_{i}, \text{subteam}(\tau_{i}, \tau'_{1})) \land \text{MBEL}(\tau_{i}, \text{subteam}(\tau_{i}, \tau'_{2})) \land \ldots \text{MBEL}(\tau_{i}, \text{subteam}(\tau_{o}, \tau'_{v})) \land \neg \text{MBEL}(\tau_{i}, \psi_{k})) \{\text{Propositional Reasoning}\}$

Theorem 6  The $\text{BGI}^{ct}$-system is a sound and complete axiomatization with respect to $\mathcal{M}^{ct}$.

Proof:

The proof for the joint mental attitudes is essentially identical to the proof provided in Chapter 5 for Theorem 4. The only modification is with respect to Case 3, i.e., node $n$ is an internal node of the tree and is a fully expanded propositional $\text{CTL}$ tableau. Here we do not have $\mathcal{B}a_{i}, \mathcal{G}a_{i}, \mathcal{I}a_{i}$-successors but rather $\mathcal{M}B_{\tau_{i}}, \mathcal{J}G_{\tau_{i}}, \mathcal{IJ}_{\tau_{i}}$-successors.

As to formulas of the form $\text{SUBTEAM}(\tau_{i}, \tau'_{j})$, these are considered as two formulas of the form $\text{MBEL}(\tau_{i}, \text{subteam}(\tau_{i}, \tau'_{j}))$ and $\text{MBEL}(\tau_{j}, \text{subteam}(\tau_{i}, \tau'_{j}))$. They are thus covered by the proof of the mutual belief formulas and accessibility relations. That is, the inconsistency

© Copyright Gil Tidhar, 1999 103
in the case of a \textsc{Subteam} formula will arise with respect to the mutual beliefs about the \textit{subteam} proposition.

Note that the proof of soundness and completeness is actually based on the pseudo-tableau construction and the inconsistency lemmas. Also note that we have changed the pseudo-tableau construction algorithm in a way that corresponds to the changes above. It follows that the proof for soundness and completeness for the joint mental attitudes is identical for the proof in the multi-agent case.

\[\blacksquare\]
Chapter 7

Axiom Systems for Team-Oriented Systems

The model described in the previous section describes a basic model of a team-oriented system. One can extend this model by adding a variety of axiom systems that constrain the relationship between the mental attitudes of a team and the mental attitudes of the sub-teams. The choice of the constraints will depend on the way these mental attitudes are to be used in the coordination of the team’s activity. Fagin et. al. [39] provide a detailed discussion of the role of similar mental attitudes in the coordination of a group of agents. In particular they focus on the role of “common knowledge” in a variety of types of distributed systems. Here we show that one can restrict the team mental attitudes to hold if and only if:

1. the same mental attitudes are held by all the sub-team; or

2. the same mental attitude is held by all the sub-teams and the team mental attitude is common knowledge for the sub-teams.\(^1\)

The above two examples of constraints on the mental attitudes of the team and the mental attitudes of the sub-teams are similar to the modalities \(E_G\) (“every agent in \(G\) knows that”) and \(C_G\) (“it is common knowledge for the agents in \(G\) that”) as defined by Halpern and Moses [54].

We represent the above two types of constraints as particular axioms in a team-oriented system. We refer to these systems as \(BG_{\text{CTL}}^{\text{tos-e}}\) and \(BG_{\text{CTL}}^{\text{tos-c}}\) respectively. A detailed comparison between our approach and the approach of Halpern and Moses is provided in the following sections.

\(^1\)Note that this approach is self referential.
7.1 Mutual Beliefs, Joint Goals, and Joint Intentions for Every Sub-team

Recall that in the $\text{BGITL}^{\text{os}}$ model we have provided for each team $\tau$ in the set of teams $T_M$ the three modalities $\mathcal{MB}_\tau$, $\mathcal{JG}_\tau$, $\mathcal{JI}_\tau$. Also recall that we refer to a team with no sub-teams as an agent or a primitive team. One can define a system in which the behavior of the team is composed of the coordinated behavior of its sub-teams. The problem presented is how to facilitate the specification of such coordinated behavior. As suggested above one can use the joint mental attitudes as the means for such a specification. The question is “How?”

One approach to the coordination of a team composed of a set of sub-teams is to require that the joint mental attitudes of the team be shared by the sub-teams. We thus ensure that the joint mental attitudes of the team are internalized by the sub-teams. For example, if a team $\tau$ has a mutual belief $\phi$ (denoted $\text{MBEL}(\tau, \phi)$) the this implies that every sub-team $\tau_i$ of $\tau$ also has this mutual belief (denoted $\text{MBEL}(\tau_i, \phi)$), and similarly for joint goals and joint intentions. It will follow that the sub-teams will act according to the mental attitudes of the team. At the end of section we discuss in some detail the benefits and limitations of this approach.

Before we proceed we would like to note that although the modal operator that we use here is $\text{MBEL}$ the semantic conditions that constrain it are similar to the those used in the definition of $\text{EBEL}$ or $E_G$ [55]. Nevertheless note that the definition of $\text{MBEL}$ as a modal operator for the team is independent of the $\text{MBEL}$ modalities of the sub-teams. The relationship between the modalities of the team and the modalities of the sub-teams is given as a special case of a team-oriented system. The operator $\text{EBEL}$ on the other hand is defined in-terms of the $\text{BEL}$ modalities of the agents that are members of the team. A comparison between the axiom system described here and the work of Halpern and Moses on the $E_G$ operator [55] is provided below.

7.1.1 Syntactic Conditions (Axioms)

Let us provide the set of axioms that should be added to the axioms in the $\text{BGITL}^{\text{os}}$ system. We refer to these axioms as the every sub-team axioms and formally define them as:

$$(\text{MB-EA}) \quad \text{MBEL}(\tau, \phi) \land \text{SUBTEAM}(\tau, \tau') \supset \text{MBEL}(\tau', \phi);$$

$$(\text{JG-EA}) \quad \text{JGOAL}(\tau, \phi) \land \text{SUBTEAM}(\tau, \tau') \supset \text{JGOAL}(\tau', \phi);$$

$$(\text{JI-EA}) \quad \text{JINTEND}(\tau, \phi) \land \text{SUBTEAM}(\tau, \tau') \supset \text{JINTEND}(\tau', \phi).$$

2Given that an agent is aware of its own mental attitudes and has total control over its actions then for a single agent the problem of coordinating its activity is transformed into the problem of single agent deliberation (see Rao and Georgeff [86] for a discussion on single agent deliberation).
The BGI\textsuperscript{pos-ec}\textsubscript{CTL} system together with the above conditions will be called the BGI\textsuperscript{pos-ec}\textsubscript{CTL} system. Note that for a primitive team (i.e., an agent) the above axioms are valid in any system.

### 7.1.2 Semantic Conditions

The semantic conditions that enforce the above model are based on a relationship between the accessibility relations of the teams and the accessibility relations of the sub-teams. From a conceptual point of view we would like the team to consider all the possible worlds of all the sub-teams. This can be represented as a multi-modal containment condition. That is, the mutual-belief, joint-goal, and joint-intention-accessible worlds of the team contain the union of the respective mutual-belief, joint-goal, and joint-intention-accessible worlds of the sub-teams. We thus require that for every team $\tau$ the following rules hold:

\begin{itemize}
  \item \textbf{(MB-EC)} $\forall w \forall t \forall v$ if $(w, t, v) \in \bigcup_{ST(\tau, \tau', w, t)} MB_{\tau'}$ then $(w, t, v) \in MB_{\tau};$
  \item \textbf{(JG-EC)} $\forall w \forall t \forall v$ if $(w, t, v) \in \bigcup_{ST(\tau, \tau', w, t)} JG_{\tau'}$ then $(w, t, v) \in JG_{\tau};$
  \item \textbf{(JI-EC)} $\forall w \forall t \forall v$ if $(w, t, v) \in \bigcup_{ST(\tau, \tau', w, t)} JI_{\tau'}$ then $(w, t, v) \in JI_{\tau}.$
\end{itemize}

The tableau rules for the axioms and semantic conditions described above are as follows:

\begin{itemize}
  \item \textbf{(MB-ET)} If $MBEL(\tau, \phi) \in \Phi(w)$ and $(w, t, v) \in \bigcup_{ST(\tau, \tau', w, t)} MB_{\tau'}$ then $\forall \tau': \text{SUBTEAM}(\tau, \tau') \in \Phi(w)$ we have $MBEL(\tau', \phi) \in \Phi(w)$ and $\phi \in \Phi(v);$
  \item \textbf{(JG-ET)} If $JGOAL(\tau, \phi) \in \Phi(w)$ and $(w, t, v) \in \bigcup_{ST(\tau, \tau', w, t)} JG_{\tau'}$ then $\forall \tau': \text{SUBTEAM}(\tau, \tau') \in \Phi(w)$ we have $JGOAL(\tau', \phi) \in \Phi(w)$ and $\phi \in \Phi(v);$
  \item \textbf{(JI-ET)} If $JINTEND(\tau, \phi) \in \Phi(w)$ and $(w, t, v) \in \bigcup_{ST(\tau, \tau', w, t)} JI_{\tau'}$ then $\forall \tau': \text{SUBTEAM}(\tau, \tau') \in \Phi(w)$ we have $JINTEND(\tau', \phi) \in \Phi(w)$ and $\phi \in \Phi(v).$
\end{itemize}

Note that for every set $\Theta$ it holds that $\Theta \subseteq \Theta$. Also note that for a primitive team (i.e., an agent), $\tau$, we have that $\tau$ is the only sub-team of $\tau$. It follows that for a primitive team (i.e., an agent) the above rules are equivalent to the BGI\textsuperscript{mas}\textsuperscript{-}consistency rules.

As is the case with possible worlds, the more worlds we consider possible the less we can be certain about the world. In our case, given the team considers the union of possible worlds considered by the sub-teams$^3$ then for a given formula there are potentially less worlds in which it can mutually believe, jointly achieve, or jointly intend, this formula compared with any of its sub-teams. The intuition behind such a model is that to attain

$^3$Note that the team may consider as possible, worlds that are not considered as possible by any of the sub-teams.
the mental attitudes of a team requires the coordination of the mental attitudes of the
sub-teams. Given such a requirement, we get that the number of worlds in which such
coordination is feasible is smaller than the number of worlds in which each of the sub-
teams can hold its own mental attitudes.

7.1.3 Soundness and Completeness

The $\text{BGI}^{\text{tos-e}}_{\text{CTL}}$ axiom system and model described above are an extension to the team-
oriented model, $\text{BGI}^{\text{tos}}_{\text{CTL}}$, described in Chapter 6. Here we will modify the proof of sound-
ness and completeness for a team-oriented model to take into account the new concepts
introduced.

We can proceed by defining $\text{BGI}^{\text{tos-e}}_{\text{CTL}}$-tableau and pseudo-$\text{BGI}^{\text{tos-e}}_{\text{CTL}}$-tableau. We begin
by defining a $\text{BGI}^{\text{tos-e}}_{\text{CTL}}$-tableau:

Definition 16

A $\text{BGI}^{\text{tos-e}}_{\text{CTL}}$-tableau is a $\text{BGI}^{\text{tos}}_{\text{CTL}}$-tableau that also satisfies conditions (MB-ET),
(JG-ET), and (JI-ET).

Similarly, one can define a pseudo-$\text{BGI}^{\text{tos-e}}_{\text{CTL}}$-tableau. We refer to the class of structures
$\mathcal{M}^{\text{tos-e}}$ that satisfies the multi-modal containment conditions (MB-EC), (JG-EC), and
(JI-EC) as $\mathcal{M}^{\text{tos-e}}$. Let us now prove the small model theorem for $\text{BGI}^{\text{tos-e}}_{\text{CTL}}$.

Theorem 7 Let $\phi_0$ be a BGI$_{\text{CTL}}$ formula of length $n$, then the following equivalences
hold:

1. $\phi_0$ is $\text{BGI}^{\text{tos-e}}_{\text{CTL}}$-satisfiable.;
2. $\phi_0$ has a model $\mathcal{M}^{\text{tos-e}}$ with a finite branching in each world bounded by $O(n)$;
3. $\phi_0$ has a finite pseudo-$\text{BGI}^{\text{tos-e}}_{\text{CTL}}$-tableau of size $\leq \exp(n)$;
4. $\phi_0$ has a finite model $\mathcal{M}^{\text{tos-e}}$ of size $\leq \exp(n)$.

Proof:

For the CTL component the proof is identical to the proof of Theorem 5. For the
BGI component we prove the equivalence between the semantic constraints (MB-EC),
(JG-EC), and (JI-EC), and the tableau rules (MB-ET), (JG-ET), and (JI-ET), re-
spectively. We will proceed to show that (MB-EC) is equivalent to (MB-ET). The proof
for the other equivalences is similar.

1. For one direction, given (MB-EC), that is $\forall w \forall t \forall v$ if $(w, t, v) \in \bigcup_{\tau, \tau'} \mathcal{M}B_{\tau'}$
then $(w, t, v) \in \mathcal{M}B_{\tau}$, we want to prove that (MB-ET) holds, that is if $\text{MBEL}(\tau, \phi)$
7.1. MUTUAL BELIEFS, JOINT GOALS, AND JOINT INTENTIONS FOR EVERY

SUBTEAM

CHAPTER 7. AXIOM SYSTEMS FOR TEAM-ORIENTED SYSTEMS

\[ \in \Phi(w_t) \text{ and } (w, t, v) \in \bigcup_{\tau \in \mathcal{ST}(\tau', \tau, w, t)} MB_{\tau'} \text{ then } \forall \tau' : \text{SUBTEAM}(\tau, \tau') \in \Phi(w_t) \text{ we have } MBEL(\tau', \phi) \in \Phi(w_t) \text{ and } \phi \in \Phi(v_t). \]

We first show that \( \phi \in \Phi(v_t) \). Given that \((w, t, v) \in \bigcup_{\tau \in \mathcal{ST}(\tau', \tau, w, t)} MB_{\tau'}\), we get from the assumption that \((w, t, v) \in MB_{\tau}\). It follows that given that \(MBEL(\tau, \phi) \in \Phi(w_t)\) we get from the BGI\textsuperscript{pos}-consistency rules that \(\phi \in \Phi(v_t)\).

We now need to show that it also holds that \(\forall \tau' : \text{SUBTEAM}(\tau, \tau') \in \Phi(w_t)\) we have \(MBEL(\tau', \phi) \in \Phi(w_t)\). Without loss of generality we will consider a particular team \(\tau_k\) such that \(\text{SUBTEAM}(\tau, \tau_k) \in \Phi(w_t)\). We will assume that \(MBEL(\tau_k, \phi) \not\in \Phi(w_t)\) and show a contradiction.

Based on the assumption and the BGI\textsuperscript{pos}-consistency rules we get that there exists \(v'\) such that \((w, t, v') \in MB_{\tau_k}\) and \(\phi \notin \Phi(v_t')\).

Given that \(\text{SUBTEAM}(\tau, \tau_k) \in \Phi(w_t)\) we get from the sub-team definition that \(ST(\tau, \tau_k, w, t)\). Given that \((w, t, v') \in MB_{\tau_k}\) we get from the assumed rule that \((w, t, v') \in MB_{\tau}\).

From the first part of this proof we get that it holds that \(\phi \in \Phi(v_t')\) which contradicts our findings that \(\phi \not\in \Phi(v_t')\).

2. For the reverse direction, given \((\text{MB-ET})\), that is if \(MBEL(\tau, \phi) \in \Phi(w_t)\) and \((w, t, v) \in \bigcup_{\tau \in \mathcal{ST}(\tau', \tau, w, t)} MB_{\tau'}\) then \(\forall \tau' : \text{SUBTEAM}(\tau, \tau') \in \Phi(w_t)\) we have \(MBEL(\tau', \phi) \in \Phi(w_t)\) and \(\phi \in \Phi(v_t)\), we want to prove that \((\text{MB-EC})\) holds, that is \(\forall w \forall t \forall v\) if \((w, t, v) \in \bigcup_{\tau \in \mathcal{ST}(\tau', \tau, w, t)} MB_{\tau'}\) then \((w, t, v) \in MB_{\tau}\).

Without loss of generality we will consider a particular team \(\tau_j\), worlds \(w'\) and \(v'\), time point \(t'\), and proposition \(\phi'\). We assume that \(MBEL(\tau_j, \phi') \in \Phi(w_{t'})\) and \((w', t', v') \in \bigcup_{\tau \in \mathcal{ST}(\tau_j, \tau', w', t')} MB_{\tau'}\). It follows that \(\forall \tau' : \text{SUBTEAM}(\tau_j, \tau') \in \Phi(w_{t'})\) we have \(MBEL(\tau', \phi') \in \Phi(w_{t'})\) and also that \(\phi' \in \Phi(v_{t'})\). We need to show that \((w', t', v') \in MB_{\tau_j}\). We will assume that \((w', t', v') \notin MB_{\tau_j}\) and show a contradiction.

If \((w', t', v') \notin MB_{\tau_j}\) then \(MBEL_{\tau_j}([w_{t'}]) \not\subseteq [v_{t'}]\). In other words, there exists \(\psi\) such that \(\psi \in MBEL_{\tau_j}([w_{t'}]) \text{ and } \psi \not\in [v_{t'}]\).

As \((w', t', v') \in \bigcup_{\tau \in \mathcal{ST}(\tau_j, \tau', w', t')} MB_{\tau'}\), there exists a team \(\tau_k\) such that \(ST(\tau_j, \tau_k, w', t')\) holds, and for \(\tau_k\) we have \((w', t', v') \in MB_{\tau_k}\) and \(MBEL_{\tau_k}([w_{t'}]) \subseteq [v_{t'}]\). Given that \(\psi \in MBEL_{\tau_j}([w_{t'}])\) then from the definition we get that \(MBEL(\tau_j, \psi) \in \Phi(w_{t'})\) and from our assumption we get that for \(\tau_k\) we have \(MBEL(\tau_k, \psi) \in \Phi(w_{t'})\). It follows that \(\psi \in MBEL_{\tau_k}([w_{t'}])\). Given that \(MBEL_{\tau_k}([w_{t'}]) \subseteq [v_{t'}]\) we get that \(\psi \in [v_{t'}]\), which contradicts our findings that \(\psi \not\in [v_{t'}]\).

Let us now describe the algorithm for constructing a pseudo-BGI\textsuperscript{pos-e}-tableau. Again the algorithm is essentially the same algorithm described in Chapter 6 with additional steps added after Step 2(d) to take into account the additional constraints. These steps ensure that when a node includes a joint mental attitude of a team then the relevant nodes are
Step 2(d):

M0: ¬MBEL(b,p), MBEL(a,p), SUBTEAM(a,b)

M1: subteam(a,b), ¬p

Step 2(d ‘):

M0: ¬MBEL(b,p), MBEL(a,p), SUBTEAM(a,b)

M1: subteam(a,b), ¬p,p

Figure 7.1: An example of a pseudo-BGI\textsuperscript{itos\_CTL} -tableau.

added to ensure that the respective joint mental attitudes of the sub-teams are considered.

The additional step, referred to as 2(d’) is as follows:

2(d’)

1. if \( \Phi(w_t) \) contains \( \text{MBEL}(\tau, \phi) \) then \( \forall \tau’ \) such that \( \Phi(w_t) \) contains \( \text{SUBTEAM}(\tau, \tau’) \), then let \( \Phi(w_t) = \Phi(w_t) \cup \{ \text{MBEL}(\tau’, \phi) \} \) and \( \forall v \) such that \( (w, t, v) \in \mathcal{MB}_{\tau’} \), let \( \Phi(v_t) = \Phi(v_t) \cup \{ \phi \} \). If there is no such \( v \) then create one and initialize as above.

2. if \( \Phi(w_t) \) contains \( \text{JGOAL}(\tau, \phi) \) then \( \forall \tau’ \) such that \( \Phi(w_t) \) contains \( \text{SUBTEAM}(\tau, \tau’) \), then let \( \Phi(w_t) = \Phi(w_t) \cup \{ \text{JGOAL}(\tau’, \phi) \} \) and \( \forall v \) such that \( (w, t, v) \in \mathcal{JG}_{\tau’} \), let \( \Phi(v_t) = \Phi(v_t) \cup \{ \phi \} \). If there is no such \( v \) then create one and initialize as above.

3. if \( \Phi(w_t) \) contains \( \text{JINTEND}(\tau, \phi) \) then \( \forall \tau’ \) such that \( \Phi(w_t) \) contains \( \text{SUBTEAM}(\tau, \tau’) \), then let \( \Phi(w_t) = \Phi(w_t) \cup \{ \text{JINTEND}(\tau’, \phi) \} \) and \( \forall v \) such that \( (w, t, v) \in \mathcal{JI}_{\tau’} \), let \( \Phi(v_t) = \Phi(v_t) \cup \{ \phi \} \). If there is no such \( v \) then create one and initialize as above.

4. If the label of a new leaf node \( w_t \) is identical to the label of an ancestral node, then erase node \( w_t \).

Note that for a given joint mental attitude of a team, we only consider the joint mental attitudes of teams that are considered sub-teams of the given team in the given world. The additional complexity involved in the above steps is thus a combination of: (1) the number of sub-team relations; and (2) the number of pairs in the team-oriented BGI modalities. Recall that the number of sub-team relations is given by the size of the set \( \mathcal{ST} \) (denoted \( |\mathcal{ST}| \)). Also, the number of pairs in the team-oriented BGI modalities is denoted \( |\mathcal{BGI}_{\text{itos}}| \).
Given the above construction we get that for a given formula $\phi$ and structure $M^{\text{tos-}e} = (\mathcal{T}, \mathcal{M}, \mathcal{E}, \mathcal{T}, \prec, \mathcal{C}, \mathcal{W}, \mathcal{ST}, \mathcal{MB}, \mathcal{JG}_i, \mathcal{JI}_i, U, \Phi)$, there is an algorithm for checking if $\phi$ is satisfied in $M^{\text{tos-}e}$ that runs in time $O((|W| + |BGI^{\text{tos}}| + (|\mathcal{ST}| \ast |BGI^{\text{tos}}|)) \ast |\phi|)$.

Consider the tableau illustrated in Figure 7.1. We start with the same formulas as in Figure 6.1 (Section 6.5). Note the effect of adding Step 2(d') - node M1 is blatantly inconsistent because it now includes both the propositions $p$ and $\neg p$. We are now in a position to prove that soundness and completeness of the $BGI^{\text{tos-}e}$-system with respect to $\mathcal{M}^{\text{tos-}e}$.

**Theorem 8** The $BGI^{\text{tos-}e}_{\mathcal{CTL}}$-system is a sound and complete axiomatization with respect to $\mathcal{M}^{\text{tos-}e}$.

**Proof:**

To show that $BGI^{\text{tos-}e}_{\mathcal{CTL}}$-system is sound and complete we need to show that a formula $\phi_0$ is $BGI^{\text{tos-}e}_{\mathcal{CTL}}$-provable if and only if $\phi_0$ is marked ‘satisfiable’ in a pseudo-$BGI^{\text{tos-}e}_{\mathcal{CTL}}$-tableau. The proof is essentially identical to the proof provided in Chapter 6 for Theorem 6.

We need to show that the axioms (MB-EA), (JG-EA), and (JI-EA) are equivalent to the tableau rules (MB-ET), (JG-ET), and (JI-ET) respectively. We will show that (MB-EA) is equivalent to (MB-ET). The other equivalences can be shown in a similar way.

1. For one direction, given (MB-EA), that is $\text{MBEL}(\tau, \phi) \land \text{SUBTEAM}(\tau, \tau') \supset \text{MBEL}(\tau', \phi)$, we want to prove that (MB-ET) holds, that is if $\text{MBEL}(\tau, \phi) \in \Phi(w_i)$ and $(w, t, v) \in \bigcup_{\mathcal{ST}(\tau, \tau', w, t)} \mathcal{MB}_{\tau'}$ then $\forall \tau' : \text{SUBTEAM}(\tau, \tau') \in \Phi(w_i)$ we have $\text{MBEL}(\tau', \phi) \in \Phi(w_i)$ and $\phi \in \Phi(v_i)$.

   Without loss of generality we will consider particular teams $\tau_j$ and $\tau_k$, worlds $w'$ and $v'$, and time point $t'$, it holds that $\text{MBEL}(\tau_j, \phi) \in \Phi(w'_i)$, $\mathcal{ST}(\tau_j, \tau_k, w', t')$ holds, $(w', t', v') \in \mathcal{MB}_{\tau_k}$, and show that $\text{MBEL}(\tau_k, \phi) \in \Phi(w'_i)$ and $\phi \in \Phi(v'_i)$.

   Let us first show that $\text{MBEL}(\tau_k, \phi) \in \Phi(w'_i)$. Given that $\text{MBEL}(\tau_j, \phi) \in \Phi(w'_i)$ then from the quotient construction we get that $M^q, [w'_i] \models \text{MBEL}(\tau_j, \phi)$.

   Given that we assumed that $\mathcal{ST}(\tau_j, \tau_k, w', t')$ holds, and hence $\mathcal{ST}^q(\tau_j, \tau_k, [w'_i])$ holds, we get that $\text{SUBTEAM}(\tau_j, \tau_k) \in \Phi(w'_i)$. From the quotient construction we get that $M^q, [w'_i] \models \text{SUBTEAM}(\tau_j, \tau_k)$.

   From (MB-EA) we now get that $M^q, [w'_i] \models \text{MBEL}(\tau_k, \phi)$. It follows that $\text{MBEL}(\tau_k, \phi) \in \Phi(w'_i)$ as required.

   We will now show that $\phi \in \Phi(v'_i)$. Note that we have assumed that $(w', t', v') \in \mathcal{MB}_{\tau_k}$ and that we have shown above that $\text{MBEL}(\tau_k, \phi) \in \Phi(w'_i)$. It follows from the $BGI^{\text{tos}}$-consistency rules that $\phi \in \Phi(v'_i)$ as required.

2. For the reverse direction, given (MB-ET), that is if $\text{MBEL}(\tau, \phi) \in \Phi(w_i)$ and $(w, t, v) \in \bigcup_{\mathcal{ST}(\tau, \tau', w, t)} \mathcal{MB}_{\tau'}$ then $\forall \tau' : \text{SUBTEAM}(\tau, \tau') \in \Phi(w_i)$ we have $\text{MBEL}(\tau', \phi) \in \Phi(v_i)$ as required.
7.1. MUTUAL BELIEFS, JOINT GOALS, AND JOINT INTENTIONS FOR EVERY SUB-TEAM

Chapter 7. Axiom Systems for Team-Oriented Systems

Φ(w_t) and φ ∈ Φ(v_t), we want to prove that (MB-EA) holds, that is MBEL(τ, φ) ∧ SUBTEAM(τ, τ') ⊃ MBEL(τ', φ).

Without loss of generality, we will consider particular teams τ_j and τ_k, world w', time point t', such that MBEL(τ_j, φ) and SUBTEAM(τ_j, τ_k) holds. We will show that MBEL(τ_k, φ) holds.

Given that MBEL(τ_j, φ) holds, we have that M^q, [w'_τ] |= MBEL(τ_j, φ). It follows that MBEL(τ_j, φ) ∈ Φ(w'_τ). Similarly we can show that SUBTEAM(τ_j, τ_k) ∈ Φ(w'_τ). It follows that τ_k ∈ SUBTEAM^-(τ_j, [w'_τ]) and hence ST^q(τ_j, τ_k, [w'_τ]) holds. It follows that ST(τ_j, τ_k, w', t') holds.

From the tableau rule we get that MBEL(τ_k, φ) ∈ Φ(w'_τ). It follows that M^q, [w'_τ] |= MBEL(τ_k, φ). We thus get that MBEL(τ_k, φ) holds.

7.1.4 Comparing BGI^{to-s-e}_{CTL} and Halpern and Moses’s E_G

The above BGI^{to-s-e}_{CTL} model is similar to using the E_G modal operator [55] for defining the BGI modal operators. Nevertheless there are a number of differences between the two approaches. In particular: (1) the model of a team; (2) the independence of the modal operators; (3) the definition of a world and language used; and (4) the number of accessibility relations. An additional minor difference is that here we do not consider knowledge but rather consider beliefs, goals, and intentions.

The model of a team: Halpern and Moses do not have an explicit definition of a team. Rather, they simply consider a set of agents that operate in a multi-agent system. This approach is similar to the model described by us in previous work [69]. Here we consider teams as first class entities and define a sub-team relationship between teams. Furthermore, we consider teams that have a choice of actions and deliberate about this choice. These teams also operate in a world that changes in uncertain ways.

The independence of the modal operators: Halpern and Moses define the special E_G modal operator in terms of the existing knowledge modal operators for a given set of agents G. That is, E_G is syntactic sugar for a conjunction of modal operators for the members of G. As mentioned above here we consider a first class team entity and define a sub-team relationship between the teams. We consider independent modalities for each of the teams. We then define the joint mental attitudes as relationships between the modalities for the team and the modalities for its sub-teams. This allows the designer to consider different types of team knowledge in a single framework.
7.2. JOINT MENTAL ATTITUDES AS COMMON MODALITIES (OR SELF REFERENTIAL AXIOMS)

The structure of the team: Halpern and Moses only consider teams which are sets of agents. They do not consider the possibility of teams being composed of other teams. That is, the only team structure considered is a flat structure. In the team-oriented approach we can also consider a variety of other team structures with varying levels in them. In particular the sub-teams of a team could be both agents and other teams.

The definition of a world and language used: Halpern and Moses consider a world to be a set of primitive propositions. They consider propositional logic augmented by the knowledge modal operators, one for each agent, as the language. Here we consider a world to be a time tree structure. Each node represents a set of propositions. We consider Computational Tree Logic (CTL) augmented by the mutual belief, joint goal, and joint intention modal operators, one for each team.

The number of accessibility relations: The main problem with the team-oriented approach comes from the requirement to explicitly define a large number of teams and appropriate accessibility relations for them. Although the designer can structure the system in any way required, it also means that this specification is not implicit. Rather it is explicit and as such requires effort in providing it. As in example in the Halpern and Moses approach if we have only $n$ agents then the designer has to specify only $n$ knowledge accessibility relations of the form $K_i$. All possible teams in the system will have the same type of team structure and team knowledge. The team structure will simply be the set of agents and the team knowledge will be derived from the agents’ accessibility relations. The number of possible teams that could be created would typically be much larger than $n$. The designer will be required to provide the team structure and accessibility relation for each of these teams.

7.2 Joint Mental Attitudes as Common Modalities (or Self Referential Axioms)

As mentioned above Fagin et al. [39] show that using common knowledge can be critical in coordination of distributed systems in general and multi-agent systems in particular.

The relationship between the mutual beliefs held by the team and the mutual beliefs held by the sub-teams may be represented as some conjunction of the mutual beliefs of the sub-teams participating in the team. The particular representation will depend on the particular type of joint mental attitude required for the operation of the team.

Similarly, a joint goal that a team wants to be satisfied may be defined in terms of a conjunction of joint goals and mutual beliefs about these goals held by the sub-teams participating in the team. Such a system corresponds to the approach taken by us previously [69]. A similar approach was taken by Rao et. al. in [91]. From a semantic point of view these definitions can also be given in terms of the relationships between the sets $\mathcal{MB}_\tau$, $\mathcal{JG}_\tau$, $\mathcal{JI}_\tau$ for the relevant team and sub-teams.
7.2. Joint Mental Attitudes as Common Modalities (or Self-Referential Axioms)

7.2.1 Syntactic Conditions (Axioms)

The axioms defined here use self-referential definitions to describe the notions of mutual beliefs, joint goals, and joint intentions. If a team holds a mutual belief $\phi$ then every sub-team mutually believes $\phi$ and every sub-team mutually believes that the mutual belief is held by the team. If a team has a joint goal towards $\phi$ (or joint intention towards $\phi$) then every sub-team has the joint goal towards $\phi$ (or joint intention towards $\phi$) and every sub-team mutually believes that the joint goal towards $\phi$ (or joint intention towards $\phi$) is held by the team. We thus require that for every team $\tau$ the following axioms hold:

- **(MB-CA)** $\text{MBEL}(\tau, \phi) \land \text{SUBTEAM}(\tau, \tau') \supset \text{MBEL}(\tau', \phi) \land \text{MBEL}(\tau, \text{MBEL}(\tau, \phi))$;
- **(JG-CA)** $\text{JGOAL}(\tau, \phi) \land \text{SUBTEAM}(\tau, \tau') \supset \text{JGOAL}(\tau', \phi) \land \text{MBEL}(\tau', \text{JGOAL}(\tau, \phi))$;
- **(JI-CA)** $\text{JINTEND}(\tau, \phi) \land \text{SUBTEAM}(\tau, \tau') \supset \text{JINTEND}(\tau', \phi) \land \text{MBEL}(\tau', \text{JINTEND}(\tau, \phi))$.

Note that in the case of a primitive team (i.e., an agent) the above axioms can be derived from the introspection axioms. The $\text{BGI}_{\text{CTL}}$ system together with the above conditions will be called the $\text{BGI}_{\text{CTL}}$ system. This will be shown in the following sections.

7.2.2 Semantic Conditions

The semantic conditions that enforce the above model are again based on a relationship between the accessibility relations of the teams and the accessibility relations of the sub-teams. From a conceptual point of view we would require the team to first consider all the possible worlds of all the sub-teams for the given modality. Furthermore we would require the team to be aware of such considerations in all these worlds. That is, we would like the team to hold a mutual belief about the existence of the whole team’s particular joint mental attitude. Again this can be represented as a multi-modal containment condition.

- **(MB-CC)** $\forall w \forall t \forall v \forall u$ if $\text{ST}(\tau, \tau', w, t)$ and if either $(w, t, v) \in \text{MB}_{\tau'}$ or both $(w, t, v) \in \text{MB}_{\tau'}$ and $(v, t, u) \in \text{MB}_{\tau}$ then $(w, t, v)$ and $(w, t, u) \in \text{MB}_{\tau}$;
- **(JG-CC)** $\forall w \forall t \forall v \forall u$ if $\text{ST}(\tau, \tau', w, t)$ and if either $(w, t, v) \in \text{JG}_{\tau'}$ or both $(w, t, v) \in \text{MB}_{\tau'}$ and $(v, t, u) \in \text{JG}_{\tau}$ then $(w, t, v)$ and $(w, t, u) \in \text{JG}_{\tau}$;
- **(JI-CC)** $\forall w \forall t \forall v \forall u$ if $\text{ST}(\tau, \tau', w, t)$ and if either $(w, t, v) \in \text{JI}_{\tau'}$ or both $(w, t, v) \in \text{MB}_{\tau'}$ and $(v, t, u) \in \text{JI}_{\tau}$ then $(w, t, v)$ and $(w, t, u) \in \text{JI}_{\tau}$.

The tableau rules for the axioms, (MB-CA), (JG-CA), and (JI-CA), and semantic conditions, (MB-CC), (JG-CC), and (JI-CC), described above are as follows:

- **(MB-CT)** If $\text{MBEL}(\tau, \phi) \in \Phi(w_i)$ and $(w, t, v) \in \bigcup_{\text{ST}(\tau, \tau', w, t)} \{ (w, t, u) : (w, t, v) \in \text{MB}_{\tau'} \land (v, t, u) \in \text{MB}_{\tau} \}$ then...
7.2. JOINT MENTAL ATTITUDES AS COMMON MODALITIES (OR SELF REFERENTIAL AXIOMS)

7. AXIOM SYSTEMS FOR TEAM-ORIENTED SYSTEMS

∀τ′: SUBTEAM(τ, τ′) ∈ Φ(w_t) we have MBEL(τ′, φ) ∈ Φ(w_t), MBEL(τ′, MBEL(τ, φ)) ∈ Φ(w_t), and φ ∈ Φ(v_t);

(JG-CT) If JGOAL(τ, φ) ∈ Φ(w_t) and (w, t, v) ∈ ∪_{ST(τ, τ′, w, t)} (JG_τ′ ∪ \{(w, t, u) : (w, t, v) ∈ MB_τ′ and (v, t, u) ∈ JG_τ\}) then
∀τ′: SUBTEAM(τ, τ′) ∈ Φ(w_t) we have JGOAL(τ′, φ) ∈ Φ(w_t), MBEL(τ′, JGOAL(τ, φ)) ∈ Φ(w_t), and φ ∈ Φ(v_t);

(JI-CT) If JINTEND(τ, φ) ∈ Φ(w_t) and (w, t, v) ∈ ∪_{ST(τ, τ′, w, t)} (JI_τ′ ∪ \{(w, t, u) : (w, t, v) ∈ MB_τ′ and (v, t, u) ∈ JI_τ\}) then
∀τ′: SUBTEAM(τ, τ′) ∈ Φ(w_t) we have JINTEND(τ′, φ) ∈ Φ(w_t), MBEL(τ′, JINTEND(τ, φ)) ∈ Φ(w_t), and φ ∈ Φ(v_t).

7.2.3 Soundness and Completeness

The BGI^{tos-c} axiom system and model described above are an extension to the team-oriented model, BGI^{tos} CTL, described in Chapter 6. Here we will modify the proof of soundness and completeness for a team-oriented model to take into account the new concepts introduced.

We can proceed by defining BGI^{tos-c} CTL-tableau and pseudo-BGI^{tos-c} CTL-tableau. We begin by defining a BGI^{tos-c} CTL-tableau:

Definition 17
A BGI^{tos-c} CTL-tableau is a BGI^{tos} CTL-tableau that also satisfies conditions (MB-CT), (JG-CT), and (JI-CT).

Similarly, one can define a pseudo-BGI^{tos-c} CTL-tableau. We refer to the class of structures \( M^{tos-c} \) that satisfies the multi-modal containment conditions (MB-CC), (JG-CC), and (JI-CC) as \( M^{tos-c} \). Let us now prove the small model theorem for BGI^{tos-c} CTL.

Theorem 9 Let \( φ_0 \) be a BGI^{tos-c} CTL formula of length \( n \), then the following equivalences hold:

1. \( φ_0 \) is BGI^{tos-c} CTL-satisfiable;
2. \( φ_0 \) has a model \( M^{tos-c} \) with a finite branching in each world bounded by \( O(n) \);
3. \( φ_0 \) has a finite pseudo-BGI^{tos-c} CTL-tableau of size \( \leq \exp(n) \);
4. \( φ_0 \) has a finite model \( M^{tos-c} \) of size \( \leq \exp(n) \).
Proof:
For the CTL component the proof is identical to the proof of Theorem 5. For the BGI component we prove the equivalence between the semantic constraints (MB-CC), (JG-CC), and (JI-CC), and the tableau rules (MB-CT), (JG-CT), and (JI-CT), respectively. We will proceed to show that (MB-CC) is equivalent to (MB-CT). The proof for the other equivalences is similar.

1. For one direction, given (MB-CC), that is, \( \forall w \forall t \forall v \forall u \text{ if } ST(\tau, \tau', w, t) \) and if either \( (w, t, v) \in MB_{\tau'} \) or both \( (w, t, v) \in MB_{\tau'} \) and \( (v, t, u) \in MB_{\tau} \) then \( (w, t, v) \) and \( (w, t, u) \in MB_{\tau} \), we want to prove that (MB-CT) holds, that is, if \( \text{MBEL}(\tau, \phi) \in \Phi(w) \) and \( (w, t, v) \in MB_{\tau'} \cup \{(w, t, u) : (w, t, v) \in MB_{\tau'} \text{ and } (v, t, u) \in MB_{\tau}\} \) then \( \forall \tau' : \text{SUBTEAM}(\tau, \tau') \in \Phi(w) \) we have \( \text{MBEL}(\tau, \phi) \in \Phi(w), \text{MBEL}(\tau', \phi) \in \Phi(w), \) and \( \phi \in \Phi(v) \).

Let \( w' \) and \( v' \) be worlds, \( t' \) be a time point, and \( \tau_j \) be a team. We will assume that \( \text{MBEL}(\tau_j, \phi) \in \Phi(w') \) and \( (w', t', v') \in MB_{\tau_j} \cup \{(w', t', u) : (w', t', v') \in MB_{\tau_j} \text{ and } (v', t', u) \in MB_{\tau_j}\} \) and show that \( \forall \tau' : \text{SUBTEAM}(\tau_j, \tau') \in \Phi(w', \phi) \) we have \( \text{MBEL}(\tau', \phi) \in \Phi(w'), \text{MBEL}(\tau_j, \phi) \in \Phi(w'), \) and \( \phi \in \Phi(v'). \)

Let \( \tau_k \) be a team such that \( \text{SUBTEAM}(\tau_j, \tau_k) \in \Phi(w') \). We will show that \( \text{MBEL}(\tau_k, \phi) \in \Phi(w'), \text{MBEL}(\tau_k, \phi) \in \Phi(w'), \) and \( \phi \in \Phi(v'). \)

\( \text{MBEL}(\tau_k, \phi) \in \Phi(w') \): It is sufficient to show that in the quotient construction we have \( M^q, [w'_j] \models \text{MBEL}(\tau_k, \phi) \). By definition we just need to show that \( \forall v : (w', t', v) \in MB^q_{\tau_k} \) we have \( M^q, [v'_j] \models \phi \). We will assume that exists \( v' \) such that \( (w', t', v') \in MB^q_{\tau_k} \) and that \( M^q, [v'] \not\models \phi \) and we will show a contradiction.

Given that \( \text{SUBTEAM}(\tau_j, \tau_k) \in \Phi(w') \) we have by the quotient construction that \( M^q, [w'_j] \models \text{SUBTEAM}(\tau_j, \tau_k) \). By definition we thus have that \( ST^q(\tau_j, \tau_k, w', t') \) holds. Given that \( (w', t', v') \in MB^q_{\tau_j} \) then by (MB-CC) we have that \( (w', t', v') \in MB^q_{\tau_j} \).

We have assumed that \( \text{MBEL}(\tau_j, \phi) \in \Phi(w') \) we have by the quotient construction that \( M^q, [w'_j] \models \text{MBEL}(\tau_j, \phi) \). By definition we have that \( \forall v : (w', t', v) \in MB^q_{\tau_j} \) we have \( M^q, [v'_j] \models \phi \). In particular we have for \( v' \) that \( M^q, [v'_j] \models \phi \) and we have a contradiction.

\( \text{MBEL}(\tau_j \cap \text{MBEL}(\tau_j, \phi)) \in \Phi(w') \): It is sufficient to show that in the quotient construction we have \( M^q, [w'_j] \models \text{MBEL}(\tau_j \cap \text{MBEL}(\tau_j, \phi)) \). By definition we just need to show that \( \forall v : (w', t', v) \in MB^q_{\tau_k} \) we have \( M^q, [v'_j] \models \text{MBEL}(\tau_j \cap \text{MBEL}(\tau_j, \phi)) \). By definition we just need to show that \( \forall u : (v, t', u) \in MB^q_{\tau_j} \) we have \( M^q, [u'_j] \models \phi \). We will assume that exists \( v' \) such that \( (w', t', v') \in MB^q_{\tau_j} \) and \( (v', t', u') \in MB^q_{\tau_j} \) and that \( M^q, [u'_j] \not\models \phi \) and we will show a contradiction.

Given that \( \text{SUBTEAM}(\tau_j, \tau_k) \in \Phi(w') \) we have by the quotient construction that \( M^q, [w'_j] \models \text{SUBTEAM}(\tau_j, \tau_k) \). By definition we thus have that \( ST^q(\tau_j, \tau_k, w', t') \)
holds. Given that \((w', t', v') \in \mathcal{MB}_{\tau_k}^q\) and \((v', t', u') \in \mathcal{MB}_{\tau_j}^q\) then by (MB-CC) we have that \((w', t', u') \in \mathcal{MB}_{\tau_j}^q\).

We have assumed that \(\text{MBEL}(\tau_j, \phi) \in \Phi(w'_\tau)\) we have by the quotient construction that \(M^q, [w'_\tau] \models \text{MBEL}(\tau_j, \phi)\). By definition we have that \(\forall v : (w', t', v) \in \mathcal{MB}_{\tau_j}^q\) we have \(M^q, [v]_\tau \models \phi\). In particular we have for \(u'\) that \(M^q, [u']_\tau \models \phi\) and we have a contradiction.

\(\phi \in \Phi(v'_\tau)\): The first two parts of this proof have shown that for \((w', t', v') \in \bigcup_{\tau \in \mathcal{ST}(\tau_j, \tau', w', v')} (\mathcal{MB}_{\tau'}^q \cup \{(w', t', u) : (w, t, u) \in \mathcal{MB}_{\tau'}^q\text{ and } (v', t', u) \in \mathcal{MB}_{\tau_j}^q\})\) we have \(M^q, [v'_\tau] \models \phi\). By definition we get that \(\phi \in \Phi(v'_\tau)\) as required.

2. For the reverse direction, given (MB-ET), that is, if \(\text{MBEL}(\tau, \phi) \in \Phi(w_u)\) and \((w, t, v) \in \bigcup_{\tau \in \mathcal{ST}(\tau, \tau', w, t)} (\mathcal{MB}_{\tau'}^q \cup \{(w, t, u) : (w, t, u) \in \mathcal{MB}_{\tau'}^q\text{ and } (v, t, u) \in \mathcal{MB}_{\tau}^q\})\) then \(\forall \tau' : \text{SUBTEAM}(\tau, \tau') \in \Phi(w_u)\) we have \(\text{MBEL}(\tau', \phi) \in \Phi(w_u)\), \(\text{MBEL}(\tau', \text{MBEL}(\tau, \phi)) \in \Phi(w_u)\), and \(\phi \in \Phi(v'_u)\). We want to prove that (MB-EC) holds, that is, if \(\forall w' \forall \forall \forall u\) if \(\text{ST}(\tau, \tau', w, t)\) and if either \((w, t, v) \in \mathcal{MB}_{\tau}^q\) or both \((w, t, v) \in \mathcal{MB}_{\tau}^q\) and \((v, t, u) \in \mathcal{MB}_{\tau}^q\) then \((w, t, v) \in \mathcal{MB}_{\tau}^q\).

\((w, t, v) \in \mathcal{MB}_{\tau}^q\): Let \(w'\) and \(v'\) be worlds, \(t'\) be a time point, and \(\tau_j\) and \(\tau_k\) be teams. We will assume that \(\text{ST}(\tau_j, \tau_k, w', t')\) and that \((w', t', v') \in \mathcal{MB}_{\tau_k}^q\) holds and show that \((w', t', v') \in \mathcal{MB}_{\tau_j}^q\). We will assume that \((w', t', v') \notin \mathcal{MB}_{\tau_j}^q\) and show a contradiction.

If \((w', t', v') \notin \mathcal{MB}_{\tau_j}^q\) then \(\text{MBEL}_{\tau_j}^q([w'_\tau]) \subsetneq [v'_\tau]\). In other words, there exists \(\phi\) such that \(\psi \in \text{MBEL}_{\tau_j}^q([w'_\tau])\) and \(\psi \notin [v'_\tau]\).

Given that we have \((w', t', v') \in \mathcal{MB}_{\tau_k}^q\) then \(\text{MBEL}_{\tau_k}^q([w'_\tau]) \subseteq [v'_\tau]\). Given that \(\psi \in \text{MBEL}_{\tau_k}^q([w'_\tau])\) then from the definition we get that \(\text{MBEL}(\tau_j, \psi) \in \Phi(w'_\tau)\) and from (MB-CT) we get that for \(\tau_k\) we have \(\text{MBEL}(\tau_k, \psi) \in \Phi(w'_\tau)\). It follows that \(\psi \in \text{MBEL}_{\tau_k}^q([w'_\tau])\). Given that \(\text{MBEL}_{\tau_k}^q([w'_\tau]) \subseteq [v'_\tau]\) we get that \(\psi \in [v'_\tau]\), which contradicts our findings that \(\psi \notin [v'_\tau]\).

\((w, t, u) \in \mathcal{MB}_{\tau}^q\): Let \(w'\), \(v'\), and \(u'\) be worlds, \(t'\) be a time point, and \(\tau_j\) and \(\tau_k\) be teams. We will assume that \(\text{ST}(\tau_j, \tau_k, w', t')\), that \((w', t', v') \in \mathcal{MB}_{\tau_k}^q\), and that \((v', t', u') \in \mathcal{MB}_{\tau_j}^q\) holds and show that \((w', t', u') \in \mathcal{MB}_{\tau_j}^q\). We will assume that \((w', t', u') \notin \mathcal{MB}_{\tau_j}^q\) and show a contradiction.

If \((w', t', u') \notin \mathcal{MB}_{\tau_j}^q\) then \(\text{MBEL}_{\tau_j}^q([w'_\tau]) \subsetneq [u'_\tau]\). In other words, there exists \(\phi\) such that \(\psi \in \text{MBEL}_{\tau_j}^q([w'_\tau])\) and \(\psi \notin [u'_\tau]\).

Given that we have \((w', t', v') \in \mathcal{MB}_{\tau_j}^q\) then it follows that \(M^q, [v'_\tau] \not\models \text{MBEL}(\tau_j, \psi)\). That is, \(\text{MBEL}(\tau_j, \psi) \notin [v'_\tau]\).

Given that \(\psi \in \text{MBEL}_{\tau_j}^q([w'_\tau])\) then from the definition we get that \(\text{MBEL}(\tau_j, \psi) \in \Phi(w'_\tau)\) and from (MB-CT) we get that for \(\tau_k\) we have \(\text{MBEL}(\tau_k, \text{MBEL}(\tau_j, \psi)) \in \Phi(w'_\tau)\). It follows that \(\text{MBEL}(\tau_j, \psi) \in \text{MBEL}_{\tau_k}^q([w'_\tau])\). Given that \(\text{MBEL}_{\tau_k}^q([w'_\tau])\)
7.2. JOINT MENTAL ATTITUDES AS COMMON MODALITIES (OR SELF REFERENTIAL AXIOMS)

We need to show that the axioms \( \text{MB-CA}, \text{JG-CA}, \) and \( \text{JI-CA} \) are equivalent to the tableau rules \( \text{MB-CT}, \text{JG-CT}, \) and \( \text{JI-CT} \) respectively. We will show that \( \text{MB-CA} \) is equivalent to \( \text{MB-CT} \). The other equivalences can be shown in a similar way.

\[ \subseteq [v'] \] we get that \( \text{MBEL}(\tau, \psi) \in [v'] \), which contradicts our findings that \( \text{MBEL}(\tau, \psi) \not\in [v'] \).

\[ \begin{align*}
\text{2(d')} \hfill \\
1. \text{if } \Phi(w_t) \text{ contains } \text{MBEL}(\tau, \phi) \text{ then } \forall \tau' \text{ such that } \Phi(w_t) \text{ contains } \text{SUBTEAM}(\tau, \tau'), \text{ then let } \Phi(w_t) = \Phi(w_t) \cup \{ \text{MBEL}(\tau', \phi), \text{MBEL}(\tau', \text{MBEL}(\tau, \phi)) \} \text{ and } \forall v \text{ such that } (w, t, v) \\
\text{2. if } \Phi(w_t) \text{ contains } \text{JGOAL}(\tau, \phi) \text{ then } \forall \tau' \text{ such that } \Phi(w_t) \text{ contains } \text{SUBTEAM}(\tau, \tau'), \text{ then let } \Phi(w_t) = \Phi(w_t) \cup \{ \text{JGOAL}(\tau', \phi), \text{MBEL}(\tau', \text{JGOAL}(\tau, \phi)) \} \text{ and } \forall v \text{ such that } (w, t, v) \\
\text{3. if } \Phi(w_t) \text{ contains } \text{JINTEND}(\tau, \phi) \text{ then } \forall \tau' \text{ such that } \Phi(w_t) \text{ contains } \text{SUBTEAM}(\tau, \tau'), \text{ then let } \Phi(w_t) = \Phi(w_t) \cup \{ \text{JINTEND}(\tau', \phi), \text{MBEL}(\tau', \text{JINTEND}(\tau, \phi)) \} \text{ and } \forall v \text{ such that } (w, t, v) \\
\text{4. If the label of a new leaf node } w_t \text{ is identical to the label of an ancestral node, then erase node } w_t.
\end{align*} \]

We are now in a position to prove that soundness and completeness of the \( \text{BGI}_{\text{CTL}}^{\text{itos-c}} \)-system with respect to \( \mathcal{M}^{\text{itos-c}} \).

**Theorem 10** The \( \text{BGI}_{\text{CTL}}^{\text{itos-c}} \)-system is a sound and complete axiomatization with respect to \( \mathcal{M}^{\text{itos-c}} \).

**Proof:**

To show that \( \text{BGI}_{\text{CTL}}^{\text{itos-c}} \)-system is sound and complete we need to show that a formula \( \phi_0 \) is \( \text{BGI}_{\text{CTL}}^{\text{itos-c}} \)-provable if and only if \( \phi_0 \) is marked ‘satisfiable’ in a pseudo-\( \text{BGI}_{\text{CTL}}^{\text{itos-c}} \)-tableau. The proof is essentially identical to the proof provided in Chapter 6 for Theorem 6.

We need to show that the axioms \( \text{MB-CA}, \text{JG-CA}, \) and \( \text{JI-CA} \) are equivalent to the tableau rules \( \text{MB-CT}, \text{JG-CT}, \) and \( \text{JI-CT} \) respectively. We will show that \( \text{MB-CA} \) is equivalent to \( \text{MB-CT} \). The other equivalences can be shown in a similar way.
7.2. JOINT MENTAL ATTITUDES AS COMMON MODALITIES (OR SELF REFERENTIAL AXIOMS)

1. For one direction, given (MB-CA), that is, $\text{MBEL}(\tau, \phi) \land \text{SUBTEAM}(\tau, \tau') \supset \text{MBEL}(\tau', \phi)$ \land\ $\text{MBEL}(\tau', \phi)$, we want to prove that (MB-CT) holds, that is, if $\text{MBEL}(\tau, \phi) \in \Phi(w_i)$ and $(w, t, v) \in \bigcup_{\mathcal{S}T(\tau, \tau', w, t)} \{ \mathcal{M}B_{\tau'} \cup \{(w, t, u) : (w, t, v) \in \mathcal{M}B_{\tau'} \text{ and } (v, t, u) \in \mathcal{M}B_T \} \}$ then $\forall \tau' : \text{SUBTEAM}(\tau, \tau') \in \Phi(w_i)$ we have $\text{MBEL}(\tau', \phi) \in \Phi(w_i)$, $\text{MBEL}(\tau', \phi) \in \Phi(w_i)$, and $\phi \in \Phi(v_i)$.

Let $w'$ and $v'$ be worlds, $t'$ be a time point, and $\tau_j$ be a team. We will assume that $\text{MBEL}(\tau_j, \phi) \in \Phi(w'_i)$ and $(w', t', v') \in \bigcup_{\mathcal{S}T(\tau_j, \tau', w', t')}(\mathcal{M}B_{\tau'} \cup \{(w', t', u) : (w', t', v') \in \mathcal{M}B_{\tau'} \text{ and } (v', t', u) \in \mathcal{M}B_T \})$ and show that $\forall \tau' : \text{SUBTEAM}(\tau_j, \tau') \in \Phi(w'_i)$ we have $\text{MBEL}(\tau', \phi) \in \Phi(w'_i)$, $\text{MBEL}(\tau', \phi) \in \Phi(w'_i)$, and $\phi \in \Phi(v'_i)$.

$\text{MBEL}(\tau_k, \phi) \in \Phi(w'_i)$: It is sufficient to show that in the quotient construction we have $M^q, [w'_i] \models \text{MBEL}(\tau_k, \phi)$. Given that $\text{MBEL}(\tau_j, \phi) \in \Phi(w'_i)$ and $\text{SUBTEAM}(\tau_j, \tau_k) \in \Phi(w'_i)$ then from the quotient construction we have $M^q, [w'_i] \models \text{MBEL}(\tau_j, \phi) \land \text{SUBTEAM}(\tau_j, \tau_k)$. From (MB-CA) we have that $M^q, [w'_i] \models \text{MBEL}(\tau_k, \phi)$ as required.

$\text{MBEL}(\tau_k, \text{MBEL}(\tau_j, \phi)) \in \Phi(w'_i)$: It is sufficient to show that in the quotient construction we have $M^q, [w'_i] \models \text{MBEL}(\tau_k, \text{MBEL}(\tau_j, \phi))$. Given that $\text{MBEL}(\tau_j, \phi) \in \Phi(w'_i)$ and $\text{SUBTEAM}(\tau_j, \tau_k) \in \Phi(w'_i)$ then from the quotient construction we have $M^q, [w'_i] \models \text{MBEL}(\tau_j, \phi) \land \text{SUBTEAM}(\tau_j, \tau_k)$. From (MB-CA) we have that $M^q, [w'_i] \models \text{MBEL}(\tau_k, \text{MBEL}(\tau_j, \phi))$ as required.

$\phi \in \Phi(v'_i)$: The first two parts of this proof have shown that $M^q, [w'_i] \models \text{MBEL}(\tau_k, \phi)$ and $M^q, [w'_i] \models \text{MBEL}(\tau_k, \text{MBEL}(\tau_j, \phi))$. By definition we get that for $(w', t', v') \in \bigcup_{\mathcal{S}T(\tau_j, \tau', w', t')}(\mathcal{M}B_{\tau'} \cup \{(w', t', u) : (w', t', v') \in \mathcal{M}B_{\tau'} \text{ and } (v', t', u) \in \mathcal{M}B_T \})$ we have $M^q, [v'_i] \models \phi$. By definition we get that $\phi \in \Phi(v'_i)$ as required.

2. For the reverse direction, given (MB-CT), that is, if $\text{MBEL}(\tau, \phi) \in \Phi(w_i)$ and $(w, t, v) \in \bigcup_{\mathcal{S}T(\tau, \tau', w, t)} \{ \mathcal{M}B_{\tau'} \cup \{(w, t, u) : (w, t, v) \in \mathcal{M}B_{\tau'} \text{ and } (v, t, u) \in \mathcal{M}B_T \} \}$ then $\forall \tau' : \text{SUBTEAM}(\tau, \tau') \in \Phi(w_i)$ we have $\text{MBEL}(\tau', \phi) \in \Phi(w_i)$, $\text{MBEL}(\tau', \phi) \in \Phi(w_i)$, and $\phi \in \Phi(v_i)$, we want to prove that (MB-EA) holds, that is, $\text{MBEL}(\tau, \phi) \land \text{SUBTEAM}(\tau, \tau') \supset \text{MBEL}(\tau', \phi) \land \text{MBEL}(\tau', \phi)$ also hold in $w'_i$.

Let $w'$ be a world, $t'$ be a time point, and $\tau_j$ and $\tau_k$ be teams. We will assume that $\text{MBEL}(\tau_j, \phi)$ and $\text{SUBTEAM}(\tau_j, \tau_k)$ hold in $w'_i$, and show that $\text{MBEL}(\tau_k, \phi)$ and $\text{MBEL}(\tau_k, \text{MBEL}(\tau_j, \phi))$ also hold in $w'_i$.

Given that $\text{MBEL}(\tau_j, \phi)$ holds then we have that $M^q, [w'_i] \models \text{MBEL}(\tau_j, \phi)$. It follows that $\text{MBEL}(\tau_j, \phi) \in \Phi(w'_i)$. Similarly we can show that $\text{SUBTEAM}(\tau_j, \tau_k) \in \Phi(w'_i)$. From the tableau rule we get that $\text{MBEL}(\tau_k, \phi) \in \Phi(w'_i)$ and $\text{MBEL}(\tau_k, \text{MBEL}(\tau_j, \phi)) \in \Phi(w'_i)$. It follows that $M^q, [w'_i] \models \text{MBEL}(\tau_k, \phi) \land \text{MBEL}(\tau_k, \text{MBEL}(\tau_j, \phi))$. We thus get that both $\text{MBEL}(\tau_k, \phi)$ and $\text{MBEL}(\tau_k, \text{MBEL}(\tau_j, \phi))$ hold in $w'_i$ as required.
7.2.4 Comparing \( \text{BGI}_{\text{tos-c}} \) \( \text{CTL} \) and Halpern and Moses’s \( C_G \)

The above multi-modal containment conditions can also be defined using the notion of reachability between possible worlds as used by Halpern and Moses [55]. Given a set of accessibility relations we say that world \( w' \) is reachable from a world \( w \) with respect to the given set if and only if there exist a sequence of worlds starting form \( w \) and ending in \( w' \) such that each pair of worlds in the sequence is in one of the given accessibility relations. More formally:

**Definition 18**

Given a set of worlds \( W \), worlds \( w, w' \in W \), and a set of accessibility relations over \( W \), denoted \( R_1, \ldots, R_n \), we say that \( w' \) is reachable from \( w \) if there exists a sequence of worlds \( v_1, \ldots, v_m \in W \) such that \( v_1 = w \) and \( v_m = w' \), and for each world \( v_i, 1 \leq i \leq (m - 1) \), there exists an accessibility relation \( R_j, 1 \leq j \leq n \), such that \( R_j(v_i, v_{i+1}) \) holds.

Given a set of worlds \( W \) and an accessibility relation over \( W \), denoted \( R \), the set \( R^* \) denotes the set of reachable world pairs.\(^4\)

We now define the multi-modal containment constraint using the concept of reachable worlds. We thus require that the mutual-belief, joint-goal, and joint-intention-accessible worlds of the team contain every world that is reachable from the respective accessibility relations for the sub-teams. Given a team \( \tau \) we define the accessibility relation \( \text{MBU}_\tau \) to be the set \( \bigcup_{ST(\tau, \tau', w, t)} \text{MB}_\tau' \). We thus require that for every team \( \tau \) the following rule holds:

\[
\text{(MB-CC*)} \quad \forall w \forall t \forall v \forall u \text{ if } ST(\tau, \tau', w, t) \text{ and } (w, t, v) \in \text{MB}_{\tau'} \text{ and } (v, t, u) \in \text{MB}_{\tau} \text{ then } (w, t, v) \text{ and } (w, t, u) \in \text{MB}_{\tau};
\]

The condition \( \text{(MB-CC)} \) implies that the mutual-belief accessibility relation for every team is transitive. Also note that it also includes the union of the accessibility relations for the sub-teams. We thus get that mutual-belief accessibility relation for the team includes the transitive closure of the union of mutual-belief accessibility relations for the sub-teams. The equivalence between the conditions \( \text{(MB-CC)} \) and \( \text{(MB-CC*)} \) is thus a direct outcome of the transitive nature of the mutual-belief accessibility relation.

It follows that the above \( \text{BGI}_{\text{tos-c}} \) \( \text{CTL} \) model is similar to using the \( C_G \) modal operator [55] for defining the BGI modal operators. As before we have the following differences between the two approaches. In particular: (1) the model of a team; (2) the independence of the modal operators; (3) the definition of a world and language used; and (4) the number of accessibility relations. Again an additional minor difference is that here we do not consider knowledge but rather consider beliefs, goals, and intentions. If we only consider the belief

\(^4\)Note that the set \( R^* \) is also the transitive closure of the set \( R \).
operator we can easily replace it with the knowledge operator and get a model similar to that of Halpern and Moses. This comparison is provided elsewhere [131].

7.3 Introspection in a Team-Oriented Model

In classical modal logics [60] transitivity of a given modal operator, $\mathcal{L}$, is represented by axiom 4: $\mathcal{L}\phi \supset \mathcal{L}\mathcal{L}\phi$. In the context of knowledge based systems and the modality of knowledge [39] this axiom can be regarded as the introspection of the respective agent. That is “if an agent knows $\phi$ then it knows that it knows $\phi$”.

The models presented here extend the basic knowledge modalities and consider three different modalities: beliefs, goals, and intentions. Introspection would thus be considered with respect to the beliefs of the agent. That is, “if an agent has the goal to achieve $\phi$ then it believes that it has the goal to achieve $\phi$” and similarly for beliefs and intentions. We can similarly extend the notion of introspection for a team. That is, if a team has a joint goal then a team has a mutual belief that it has this joint goal, and similarly for mutual beliefs and joint intentions. We thus require that for every team $\tau$ the following axioms hold:

Team Introspection Axioms:

$$(MB-4A) \ MBEL(\tau, \phi) \supset MBEL(\tau, MBEL(\tau, \phi));$$

$$(JG-4A) \ JGOAL(\tau, \phi) \supset MBEL(\tau, JGOAL(\tau, \phi));$$

$$(JI-4A) \ JINTEND(\tau, \phi) \supset MBEL(\tau, JINTEND(\tau, \phi)).$$

The semantic conditions that enforce the above model are based on a relationship between the different accessibility relations of the teams. From a conceptual point of view we would require that the accessibility relation for the mutual-belief, joint goals, and joint intentions all include all worlds that are reachable with respect to the mutual-belief accessibility relation. More formally, we require that the following rules holds:

$$(MB-4C) \ \forall w \forall t \forall v \forall u \text{ if } (w,t,v) \in MB_{\tau} \text{ and } (v,t,u) \in MB_{\tau} \text{ then } (w,t,u) \in MB_{\tau};$$

$$(JG-4C) \ \forall w \forall t \forall v \forall u \text{ if } (w,t,v) \in MB_{\tau} \text{ and } (v,t,u) \in JG_{\tau} \text{ then } (w,t,u) \in JG_{\tau};$$

$$(JI-4C) \ \forall w \forall t \forall v \forall u \text{ if } (w,t,v) \in MB_{\tau} \text{ and } (v,t,u) \in JI_{\tau} \text{ then } (w,t,u) \in JI_{\tau}.$$  

Note that the condition $(MB-4C)$ actually implies that for every team $\tau$ the accessibility relation $MB_{\tau}$ is transitive.
7.4 Introspection in the Presence of the Every Sub-team Axioms

In this section we will describe the system resulting from combining the introspection axioms presented in Section 7.3 and the every sub-team axioms presented in Section 7.1. For the purpose of this discussion we will only consider the mutual-belief modality (i.e., MBEL). We will thus consider systems in which the axioms (MB-EA) and (MB-4A) hold. These axioms are:

\[(MB-EA) \quad MBEL(\tau, \phi) \land SUBTEAM(\tau, \tau') \supset MBEL(\tau', \phi);\]

\[(MB-4A) \quad MBEL(\tau, \phi) \supset MBEL(\tau, MBEL(\tau, \phi)).\]

For the semantic models we will consider models for which the semantic conditions (MB-EC) and (MB-4C) hold. These conditions are:

\[(MB-CC) \quad \forall w \forall t \forall v \forall u \text{ if } ST(\tau, \tau', w, t) \text{ and if } (w, t, v) \in MB_{\tau'} \text{ or if } (w, t, u) \in MB_\tau \text{ then } (w, t, v) \in MB_\tau;\]

\[(MB-4C) \quad \forall w \forall t \forall v \forall u \text{ if } (w, t, v) \in MB_\tau \text{ and } (v, t, u) \in MB_\tau \text{ then } (w, t, u) \in MB_\tau.\]

We will refer to a the above system as BGI\textsuperscript{tos-ei} CTL and to the corresponding model as M\textsuperscript{tos-ei}. In the BGI\textsuperscript{tos-ei} CTL system we have for given teams \(\tau_j\) and \(\tau_k\) and a formula \(\phi'\) the following reasoning:

1. \(MBEL(\tau_j, \phi') \land SUBTEAM(\tau_j, \tau_k)\) [assumption]
2. \(MBEL(\tau_k, \phi')\) [(1)+(MB-EA)+(MP)]
3. \(MBEL(\tau_j, MBEL(\tau_j, \phi'))\) [(1)+(MB-4A)+(MP)]
4. \(MBEL(\tau_k, MBEL(\tau_j, \phi'))\) [(3)+(MB-EA)+(MP)]
5. \(MBEL(\tau_j, \phi') \land SUBTEAM(\tau_j, \tau_k) \supset MBEL(\tau_k, \phi') \land MBEL(\tau_j, MBEL(\tau_j, \phi'))\) [(1)+(2)+(4)]

Given that this is true for any teams \(\tau_j\) and \(\tau_k\) and formula \(\phi'\) then we get that the following axiom is valid in a BGI\textsuperscript{tos-ei} CTL system:

\[(MB-EIA) \quad MBEL(\tau, \phi) \land SUBTEAM(\tau, \tau') \supset MBEL(\tau', \phi) \land MBEL(\tau', MBEL(\tau, \phi)).\]

We also get that in the M\textsuperscript{tos-ei} models for given teams \(\tau_j\) and \(\tau_k\), given worlds \(w', v', u',\) and a given time point \(t'\) the following conditions hold:

1. \(ST(\tau_j, \tau_k, w', t')\) and either \((w', t', v') \in MB_{\tau_k}\) or both \((w', t', v') \in MB_{\tau_k}\) and \((v', t', u') \in MB_{\tau_j}\) [assumption]
2. \((w', t', v') \in MB_{\tau_j}\) [(1)+(MB-EC)]
3. \((w', t'u') \in MB_{\tau_j}\) [(1)+(2)+(MB-4C)]

4. if \(ST(\tau_j, \tau_k, w', t')\) and if either \((w', t', v') \in MB_{\tau_k}\) and \((v', t', u') \in MB_{\tau_j}\) then \((w', t', v')\) and \((w', t', u') \in MB_{\tau_j}\) [(1)+(2)+(3)]

Given that this is true for any two teams, \(\tau_j\) and \(\tau_k\), worlds \(w', v', u',\) and time point \(t'\) we get that the following rule holds:

\[[MB-EIC] \forall w \forall t \forall v \forall u \text{ if } ST(\tau, \tau', w, t) \text{ and if either } (w, t, v) \in MB_{\tau'} \text{ or both } (w, t, v) \in MB_{\tau'} \text{ and } (v, t, u) \in MB_{\tau} \text{ then } (w, t, v) \text{ and } (w, t, u) \in MB_{\tau}\]

Note that the axiom and semantic condition \((MB-EIA)\) and \((MB-EIC)\) are identical to the respective axiom and semantic condition \((MB-CA)\) and \((MB-CC)\). Also, recall that we can replace the mutual-belief operator with the knowledge operator. By doing so we get that the axiom and semantic condition \((MB-CA)\) and \((MB-CC)\) are equivalent to the respective definition and semantic constraint for the common knowledge operator \(C_G\) as defined by Halpern and Moses [55].

By replacing the mutual belief operator with the knowledge operator we get the following result:

In the context of a team-oriented system the combination of every sub-team knows axiom and introspection axiom for the mutual belief of the team leads to common knowledge characteristics for this mutual belief.

To conclude, Halpern and Moses have suggested two axioms that provide different perspectives on common knowledge: (1) the infinite conjunction of nested knowledge of the members of the group; and (2) the fixed point solution to an equation involving the common knowledge operator. Our work introduces a new definition and perspective on common knowledge.

We view mutual belief (or “common belief”) as a combination of: (1) a single (finite) conjunction of sub-team belief; and (2) the introspection of the team as a whole as to its own beliefs. The infinite features of the model are not represented explicitly in the definition. Rather, they are an outcome of the interaction between the two components of the mutual belief, namely the conjunction of mutual beliefs of the members and the introspection of the team as a whole. That is, the semantic condition of \(G\)-reachability is an outcome of the mutual belief accessibility relation incorporating the union of the accessibility relations of the sub-teams and it being transitive.

This view of mutual belief has emerged from our approach of abstracting the notion of a team and mutual belief and viewing them as first class entities.

We can now consider the Halpern and Moses’ approach as a special case in which teams only have a sub-team relationship with agents. In this special case the team mutual belief operator, \(MBEL(\tau, \phi)\), will be identical to the Halpern and Moses common knowledge operator, \(C_G(\phi)\). Given that in the general approach a team can also have sub-team relationships with other teams, the team mutual belief operator is actually a generalization of the \(C_G\) operator for structured teams.
7.5 Team-Oriented Systems and Bounded Rationality

In the previous sections we have considered various team-oriented systems in the context of distributed computer systems. We described a number of systems and described how they relate to other models in computer science. In this section we would like to focus on the use of team-oriented systems in the context of organizational theory. We first describe a number of key concepts of organizational theory. In particular we focus on the concept of “bounded rationality”. We then proceed to describe how such concepts could be modelled using team-oriented systems.

7.5.1 Bounded Rationality in Organizational Theory

In a number of the theories on human organizational structures [6, 76, 77, 143] there are discussions and analysis of the size of an organization. The term “size” refers to the number of units in the organization (i.e., departments and individuals), the number of levels in the structure, and the number of units in each level in the structure.

These discussions focus on a number of related problems: (1) why do organizations grow in size; (2) why does departmentalization/specialization occur; (3) why are hierarchies formed; and (3) why is each level limited in size.

It seems that there are two key parameters that are part of any theory attempting to address these problems. These are the costs associated with achieving the desired goals and the bounded abilities of the relevant resources. The costs also include the overheads associated with establishing the team that will achieve the goals. The resources also include the humans (with their limited cognitive ability) and the communication channels (with their limitation in accuracy and reliability).

The interesting feature is that it seems that the constraints that cause organizations to emerge are primarily dynamic rather than static [143]. That is, if we consider organizations only from a static point of view it is not clear why organizations would emerge. If we consider the operations and dynamics of the organization then we see that the above limitations come into play.

In our model we do not focus on the cost or limitations of the communication channels. Nevertheless we do assume that we are dealing with computer systems with bounded computation power. We thus focus on the notion of limited cognitive ability or ”bounded rationality” [76, 110]. Limited cognitive ability leads to a number of problems that are referred to in general as problems with the “span of control” within an organization [6]. Limited cognitive ability leads to (at least) three problems:

1. the executive is limited in his/her ability to process all the information coming form the units he/she control;

2. the executive is limited in his/her ability to properly convey the organizational goals and the corresponding specific unit goals to all the unit he/she control; and
3. the executive is limited in his/her ability to properly provide the advice and information required in the process of controlling of the units he/she controls.

The number of humans in the organization and the above limitations force an organization to adopt a shape of a “pyramid”. Blau and Schoenherr [6] define the organization to be composed of a number of divisions and each division to be composed of a number of sections. Sections are internally composed of other sections. At the bottom of the hierarchy (or pyramid) are the humans that make up the different sections. Blau and Schoenherr then suggest that the shape of the pyramid is defined by three variables and their relationships: (1) the number of hierarchical levels; (2) the number of functional divisions; and (3) the number of sections per division at the agency headquarters.

We would now like to show how such concepts from organizational theory could be formalized as a specific team-oriented system. In our model we do not distinguish between divisions and sections and consider them all as different units within a team. We say that to team $\tau'$ is a unit of another team $\tau$ if there are teams $\tau_1, \ldots, \tau_n$ such that $\text{SUBTEAM}(\tau_i, \tau_{i-1})$ $2 \leq i \leq n$, $\tau_1 = \tau$, and $\tau_n = \tau'$.

We thus get that in a team-oriented formalism the shape of the pyramid is defined by two variables: (1) the number of hierarchical levels, that is, the number of sub-team relationships between the whole team and the primitive teams that are its units; and (2) the number of sub-teams for each non-primitive team.

In our discussion we will only focus on the problem of processing information. In the team-oriented formalism the processing of information corresponds to the generation of a mutual-belief given the mutual-beliefs of the sub-teams. We thus first provide the relevant axiom to state the relationship between the mutual-belief of the team and the mutual-beliefs of the sub-teams.

We will use the BGI$^{\text{os-e}}_{\text{CTL}}$ system (see Section 7.1). This system is based on the (MB-EA) axiom. The (MB-EA) axiom states that for every team, if the team mutually believes a formula then it means that each of the sub-teams must also mutually believe that formula.

As noted above, the more possible worlds we consider the less we can be certain about the world. In our case given the team considers the union of possible worlds considered by the sub-teams then for a given formula there are potentially less worlds in which it can mutually believe this formula compared with any of its sub-teams.

The intuition behind such a model is that to attain the mutual-belief of a team requires the synchronization of the mutual-beliefs of the sub-teams. Given such a requirement, we get that the number of worlds in which such synchronization is feasible is smaller than the number of worlds in which each of the sub-teams can hold its own mutual-belief. An investigation of the problems associated with the flow of information in large organizations has been conducted by Downs [30].

In the first part of the chapter we presented two models of joint mental attitudes: (1) every sub-team; and (2) common knowledge. In both models, given a set of worlds and a mutual belief, the complexity of validating that mutual belief depends on the complexity of
validating the corresponding mutual belief for each of the sub-teams. Let us now provide an analysis of the complexity of checking the validity of a mutual-belief.

7.5.2 The Complexity of Validating a Mutual Belief

In the above model we have made the mutual belief of a team imply the conjunction of the mutual beliefs of the sub-teams. We can thus define the complexity of validating a mutual belief of the team to be at least the sum of the complexity of validating the corresponding mutual beliefs for each of the sub-teams. According to the definition of a mutual belief, validating a mutual belief in a formula entails checking the validity of the formula in each of the accessible worlds of the formula. We thus get that given formula \( \phi \), a model \( M^{t_{os-e}} \), world \( w' \), time point \( t' \), team \( \tau \), and mutual belief \( \text{MBEL}(\tau, \phi) \) of \( \tau \) we have that the complexity of validating a mutual belief is:

\[
O(M^{t_{os-e}}, w' \models \text{MBEL}(\tau, \phi)) = \\
\leq O(\sum_{MB_{\tau}(w', t', v)} O(M^{t_{os-e}}, v' \models \phi)) = \\
= O(\sum_{ST(\tau, t', w', t')} (\sum_{MB_{\tau}(w', t', v)} O(M^{t_{os-e}}, v' \models \phi))).
\]

We can now repeat the same process for each of the sub-teams. This process will terminate when all sub-teams are primitive teams (i.e., agents). Note that if \( \tau \) is a primitive team then the complexity of validating a mutual-belief is \( O(|MB_{\tau}(w', t', v)| \cdot O(M^{t_{os-e}}, v' \models \phi)) \).

We will assume that for each of the primitive teams in \( M^{t_{os-e}} \) the size of the mutual-belief-accessibility relation is bound by \( K \). Furthermore we will assume that the complexity of validating \( \phi \) is bound by \( P \). We thus get that for a primitive team the complexity of validating the mutual belief is \( O(K \cdot P) \).

Given \( (MB-EC) \) we get that \( |MB_{\tau}| \leq \sum_{ST(\tau, t', w', t')} |MB_{\tau'}| \). We now define the variables \( L \) and \( S \) that correspond respectively to the bounds on: (1) the maximum number of “levels” between the team, \( \tau \), and the primitive team units; and (2) the maximum number of sub-teams for each non-primitive team unit of \( \tau \). We can now re-formulate our computation as follows:

\[
O(M^{t_{os-e}}, w' \models \text{MBEL}(\tau, \phi)) \leq \\
\leq O(S \cdot O(\sum_{MB_{\tau}(w', t', v)} O(M^{t_{os-e}}, v' \models \phi))) = \\
= O(S \cdot O(\sum_{ST(\tau, t', w', t')} (\sum_{MB_{\tau'}(w', t', v)} O(M^{t_{os-e}}, v' \models \phi)))) \leq \\
\leq O(SL \cdot (K \cdot P))
\]
Note that the two variables $L$ and $S$ are the same as the variables defining the shape of a pyramid in the context of organizational theory (see Section 7.5.1). Also note that if we have $L = 1$ then the teams that can be formed in such a team-oriented system only have primitive teams (i.e., agents) as sub-teams. This model is similar to the model developed by Halpern and Moses [55]. In that model a team is defined as a group of agents. The variable $S$ will thus correspond to the number of agents in the group. Huberman and Loch [59] verified using simulation the effects of the size of a group of artificial agents on the effectiveness of the group. In particular they show that there is a threshold up to which the increase in the size of the groups improves the effectiveness of the group. Beyond this threshold the effectiveness of the group starts to decrease.

The above formulation is a simplified version of the formulation of Williamson [143] for the accumulated production of an organization. In his formulation, in addition to the limited span of control, Williamson also considers the cost that an organization incurs. Furthermore, Williamson considers a variable number of units at each level. As mentioned above we do not consider such costs but only focus on the effect of the limited span of control.

Given that we are only interested in identifying an upper bound on our computation we simplify our formulation. We thus uniformly consider a fixed number of units per level across the organization, that is, the maximum number of units per level found in the organization. This provides us with the required upper limit.

### 7.5.3 Bounded Rationality in Team-Oriented Systems

From the formulation in the previous section it is clear that the complexity of validating a mutual-belief of a team is directly related to the sub-team relationship. If we assume that each team has limited resources to compute the validity of a joint mental attitude then we are forced to restrict the number of sub-teams that are considered at each level.

We now introduce the limitation on the processing of information or the limited span of control. As is the case in organization theory this limitation is with respect to the maximum number of units, $S$, that can be at each level of the organization. That is the maximum number of sub-teams that can realistically contribute to the creation of a mutual-belief (given the limited resources). We present the constraints on the number of sub-teams as an axiom that involves the sub-team relation.

We first define the function $n_{\text{subteams}} : T.M \rightarrow \mathbb{N}$ that returns the number of sub-teams for the given team. That is, for a given team $\tau$ the function $n_{\text{subteams}}$ returns $|\{\tau' : \text{SUBTEAM}(\tau, \tau')\}|$. Given an integer $S$ we restrict the number of sub-teams of a given team by requiring that for every team the number returned by $n_{\text{subteams}}$ is bound by $S$. More formally:

$$(\text{ST-SA}) \quad n_{\text{subteams}}(\tau) \leq S$$

A $\text{BGI}^{\text{tos-cc}}_{\text{CTL}}$ system in which the above axiom holds is referred to as a $\text{BGI}^{\text{tos-cc}}_{\text{CTL}}$ system. The semantic condition that corresponds to the above axiom is represented as a constraint
on the $ST$ relation. For a given team $\tau$, for every world and time point, the number of teams that are in a sub-team relation with $\tau$ is bound by $S$. More formally:

$$(ST-SC) \forall w \forall t \left| \{ ST(\tau, \tau', w, t) \} \right| \leq S$$

Note that the $ST$ relation is an abbreviation for a combination of mutual-belief that exist between the two teams. It is thus based on the respective mutual-belief-accessibility relations. We refer to a model in which the above condition is satisfied as $M_{tos-es}$. Given that $ST(\tau, \tau', w, t)$ iff $M_{tos, w_t} \models \text{SUBTEAM}(\tau, \tau')$ showing that $(ST-SA) \equiv (ST-SC)$ is straightforward.

Note that we have not put any constraints on the number of levels in the team, i.e., the variable $L$. We thus get that the above axiom allows only for teams that are bound by a particular sub-team relationship. Furthermore, computing a mutual-belief for the whole team becomes a function of the number of levels, $L$, in the team. In the context of Organizational Theory an investigation of the problems associated with the number of levels within an organization is provided by Downs [30].
Chapter 8

Organization-Oriented Systems

In a large team the activities of making decisions, evaluating options, and sharing information are not trivial. Furthermore, the way these are performed may affect the performance of the team and its ability to achieve its goals. The team-oriented approach described above suffers from a number of shortcomings. In particular the designer of the system cannot explicitly specify the way decisions are being made, multiple options evaluated, and information shared between sub-teams.

An organization-oriented system extends the team-oriented system with the addition of social relationships between the different components of the system. We refer to each component of a distributed system as an organization. The relationships between organizations are referred to as social mental attitudes. The relationships of interest to us are the command, control, and communication relationships.

As noted by Castelfranchi [13] these attitudes are distinct from the collective mental attitudes described in Chapter 6. Preliminary work on formalizing this approach within a BDI framework can be found in recent work by Cavedon and Tidhar [18]. Like the joint mental attitudes, social mental attitudes are with respect to a particular state of the world. Nevertheless, social mental attitudes represent a relationship between two organizations with respect to a state of the world.

Each organization includes an internal organizational model. The internal organizational model describes the organizations known to that organization and the social mental attitudes that exists between the known organizations. In particular, it includes the social mental attitudes the organization has with other organizations. A team is considered to be a special case of an organization. It is an organization with no social relationships with other organizations and with no knowledge about these relationships.
8.1 Organization-Oriented Systems: Informal Discussion

8.1.1 Background and Motivation

Recall that our main objective is to design distributed computer systems that can exhibit different types of complex behavior in a dynamic world. Throughout this work we view the design of agents, multi-agent systems, and team-oriented systems as means towards achieving this objective. When considering groups of artificial agents this underlying approach is similar to the approach taken in the development of the Systems Model of human organizations [1, 96].

Recall that in Chapter 1 we discussed the underlying assumptions for this work. We also described the type of artificial agents considered here. Like humans, the artificial agents considered here are also limited in their physical and cognitive abilities. That is, they are limited in the type and number of actions they can perform and also limited in the ability to process information, make decisions, and communicate information and decisions.

Huberman and Loch [59] have shown that for groups of artificial agents with no structure there is a threshold up to which the increase in the size of the groups without any structure improves the effectiveness of the group. Beyond this threshold the effectiveness of the group starts to decrease. The reason for this is that the overhead involved in coordinating the structureless group is very large.

Similarly, according to the systems model the reason for the emergence of human organizations with organizational structures is that they are more effective in achieving the goals of large groups. Such effectiveness is measured with respect to the time and effort that is required to achieve the goals. The gain in effectiveness is relative to the effectiveness of a structureless group. This relative gain when a structure is introduced is primarily due to the limited abilities of humans that make up the group. Because computer systems also have limited physical and cognitive abilities similar considerations are also plausible for groups of computer systems, i.e., a distributed system.

Organization Theory has been focusing on the definition and investigation of human organizations. Such definitions include different types of relationships that exist within such organizations. The types of social relationships that exist between humans are very complex. Such relationships include a variety of formal and informal relations as described in our introduction to Organization and Management Theory (Chapter 3).

The types of social relationships that we consider are associated with the operation of a group of organizations as a larger organization. Simon [108] has developed a model of decisions made by humans in an organization. He described their behavior as following formal and rational principles and as adopting a process of means-ends analysis. Here we consider organizations as first class entities with their own behavior and decision making capabilities. We adopt Simon’s approach but apply it not only to the humans (i.e., agents) in the organization but also to the organization itself.
One of the formal models of organizations and inter-organization relationships (in particular military organizations) is the Command, Control, and Communication (C3) model [56, 64, 65, 73]. Due to the critical nature of the tasks performed by military organizations the structure adopted by them has typically been relatively formally defined.

Military organizations and their structures have been documented and investigated by many historians, sociologists, psychologists, and others. Although there are many forms of military organizations, the relationships between the various components are typically relatively formal and limited in their type. The C3 model of organizational structures is to some extent a simplified model of the relationships between various components of an organization.

As the name suggests, C3 models define organizational structures in terms of the command, control, and communication relationships that exist between the various components of the organization. Despite their relative simplicity there is still no single, clear, and agreed C3 model [149]. Note again the distinction between a C3 organizational model and a C3 system. The former is a specification of an organizational structure while the later is a particular technology used in support of this structure.

When considering C3 organizational models, there are many definitions of command, control, and communication relationships [46, 95, 149]. Nevertheless there is general agreement that in some form or the other:

**Command** primarily involves the authority and responsibility to determine the objectives of the organization and the sub-units of the organization;

**Control** primarily involves the authority and responsibility for specification and modification of the detailed plans required for achieving the objectives and the monitoring of the execution of these plans; and

**Communications** primarily involves the sharing of information w.r.t the state of the environment, the state of the organization, the state of the achievement of objectives, and the state of the execution of plans.

In his work on defining the structure of organizations Mintzberg [79] states that “the parts of an organization are joined together by different flows”… [79, page 35]. These flows are identified as the operating flows and the regulating flows [79, Chapter 3]. The operating flows are related to the core operation of the organization and involve the transfer and transformation of materials and information. The regulating flows are related to the control of the core operation and involve the transfer of commands, decisions, and information.

Performance of work and coordination of this activity is specified in team (or organizational) plans. When discussing organizational structures we are primarily interested in how the performance of work is controlled. We are thus primarily interested in the regulating flows of the organization. Following is a more detailed description of the regulating flows as identified by Mintzberg [79]:

© Copyright Gil Tidhar, 1999
Authority/Command Flow: describes the set of command links that exist between the different parts of the organization. Each decision link describes the ability of one part to command and instruct another part.

Decision Flow: describes the set of decision links that exist between the different parts of the organization. Each decision link describes the ability of one part to instruct another part as to the detailed plans for achieving a command.

Performance Information Flow: describes the set of performance communication links that exist between the different parts of the organization. Each performance communication link describes the ability of one part to inform another part of the state of work and performance of a command or plan.

Staff Information Flow: describes the set of staff communication links that exist between the different staff and line managers of the organization. Each staff communication link describes the ability of staff to communicate information related to decisions to be made by line managers.

It is important to note that both commands and decisions are informational in nature. The distinction between these two flows and the two information flows is primarily with respect to the illocutionary force of the information provided. In this work we refer to each link as a social relationship. We formalize the force of the relationship by identifying the relation between the existence of the social relationship and the adoption of a joint goal or joint intention.

The model developed by Mintzberg is very elaborate and accounts for a variety of types of human organizations. In particular it is more elaborate than the theory of Command, Control, and Communication (C3) described above. For the purpose of our investigation the above description of flows corresponds to the theory of C3. Command is associated with authority flow and control is associated with the decision flow. The only distinction is that Mintzberg distinguishes between two types of communication flows (i.e., performance related communication and decision related communication). Here we consider both types of information flows as a single type of communication flow. The distinction between the two types will be significant when considering the operational semantics of the various communication flows. This will be further discussed in Chapter 10.

In this chapter we consider a theoretical model for organizational structures. We prescribe to the C3 approach and restrict our investigation to the social mental attitudes of Command, Control, and Communication. In Chapter 10 we describe the operational semantics of organizational structures.

8.1.2 Informal Discussion on The Formal Model

In the previous chapters we described mental attitudes of a single agent and joint mental attitudes of a team as a binary relation between a team (or agent) and a state of the world. This relation represents the situation in which the team (or agent) has the joint mental
attitude towards the state of the world, i.e., a belief that it is true, a goal to make it true, or an intention to fulfill such a goal.

The logical models provided did not include any aspect of the social relationships that may exist between the teams (or agents). In particular, the logical models did not include any facility for explicitly specifying the ability of a team to affect the joint mental attitudes of another team, the ability of a team to control the decisions taken by another team, or the commitment that a team has to communicate its mutual beliefs to another team.

A social mental attitude is a ternary relation between two organizations and a state of the world. This relation represents the fact that the two organizations have a social relationship with respect to the state of the world. For example, the mission commander has a commanding relationship with the sweep force with respect to intercepting a target. Note that the logical model has to be extended to include such ternary relationships.

In our discussion on team-oriented systems we also introduced the sub-team relationship to represent the structure of the team. We have described a number of particular team-oriented systems. Such systems were modelled as different axiom systems and conditions on the joint mental attitudes and team structure of the relevant teams (see Chapter 7). In a similar way, we define different types of organization-oriented systems as different axiom systems and conditions on the social mental attitudes, joint mental attitudes, and team structure of the relevant organizations.

It is important to note here that we make some assumptions that simplify the expressiveness of the model with respect to the social mental attitudes. In this work social mental attitudes are with respect to a particular state of the world. It would be a natural (but not simple) extension to have social mental attitudes refer to “abstract state of the world”, that is, a relationship with respect to a class of states. In previous work [15] we have provided some initial discussions on this issue and the benefits of using such an approach. Further investigation can be found in the work of Cavedon and Sonenberg [17].

In a team-oriented system the behavior of the team was based on its joint mental attitudes and the team structure. By introducing the social mental attitudes the behavior of an organization will now also depend on the social mental attitudes it has involving other organizations. We also have to consider the interactions between social mental attitudes and: (1) joint mental attitudes; (2) team structure; and (3) other social mental attitudes.

We consider the rationality of an organization to be one particular interaction between these three aspects of an organization. Different types of rationality are modelled using different interactions. We can thus use the organization-oriented approach for defining and investigating different types of rationality for organization-oriented systems.

An example of such a model could be one type of relationship between the command, control, and communication attitudes and the joint mental attitudes as follows:

**Command**: the authority (and responsibility) that one organization has to determine the adoption of a joint goal by another organization;

**Control**: the authority (and responsibility) that one organization has to make decisions for another organization as to the various possible ways in which the other organization manages the joint intention to achieve an adopted joint goal; and
Communication: the authority (and responsibility) that one organization has in communicating its own mutual beliefs to another organization.

Note that the above relationships are not dependent on the team-structure of the organization (represented by the sub-team relation). The social relationships can exist between a sub-team and the whole organization, e.g., a commander of a unit will decide on the current goals of the whole unit. Also, note that although the control relationship is with respect to a particular state of the world, it reflects the ability of one organization to decide on the means of achieving that state.

It is important to note that this distinction between social mental attitudes and team structure is at times unclear in the common use of organizational charts. This ambiguity is discussed by Mintzberg [79, page 36] in his discussion on the formal structure of organizations. The main ambiguity is with respect to the decomposition of organizations into divisions and sub-divisions and the authority that particular individuals within each division (i.e., the managers) have in commanding, controlling, and communicating with the sub-divisions.

Recall that an organization also has a team-structure (i.e., a set of sub-team relationships with other organizations). Given an organization, we define a social structure for it to be a set of social relationships between the sub-teams and the whole organization or between two sub-teams. The combination of team structure and social structure is regarded as the organizational structure for the organization. A formal definition of social and organizational structures can be found in Section 8.10.

8.1.3 Rationality in Organizations

The notion of “rationality” in the context of organizations is defined in Organization and Management Theory in different ways [1, 5, 75], from the rational behavior of the individual acting within an organization [110] through to the rational approach to the design of organizations [96]. In the description of the systems approach to organizations Abrahamsson [1, page 71] defines the rational model of organizations in the following way:

"The organization is viewed as an instrument, i.e., a rationally designed means for the realization of explicit goals of a particular group of people. The organizational structure is regarded as a tool, and alterations of the organizational structure are seen as instruments for improving efficiency."

What is common to all of these definitions is that rationality is a relationship between the social mental attitudes an organization has with other organizations, its joint mental attitudes, and its structure. Here we define a particular organization in which such relationship exists as a special axiom system in the organization-oriented model.

Example 1

Before we proceed, let us describe an example of the simple organizational structure as introduced by Mintzberg [79, Chapter 17]. The simple organizational structure can be
8.2. SYNTAX

CHAPTER 8. ORGANIZATION-ORIENTED SYSTEMS

typically observed in young and small organizations. Examples of this structure are an automobile dealership with a flamboyant owner, and a middle-sized retail store.

A simple organizational structure has only two prime units: the strategic apex (i.e., CEO) and the operating core (i.e., the production line). It is identified by a small management that holds both the command and control. That is, the CEO of the organization can determine the goals of the organizations and of each of the members. In C3 terms, the CEO has a command relationship with the whole organization and with every member of the organization.

Furthermore, he or she can (and in many cases will) determine for the whole organization and for each of the members of the organization how they will achieve these goals. This can be expressed as a business plan, work plan, or weekly schedule. In C3 terms the CEO has a control relationship with the organization and each of the members.

Communication links are direct between the members of the organization and the CEO. The CEO will often contact the members directly or will adopt an “open door policy” (i.e., he or she will be contacted by the employees on a regular basis). In C3 terms the CEO and the members have bi-directional communication relationships.

A formal description of a simple organizational structure is provided in Section 8.6.

8.2 Syntax

The syntax of organization-oriented systems described here extends the syntax of team-oriented systems described in Chapter 6. For the purpose of completeness we provide here the new definition of a formula in the language:

- each atomic proposition \( \phi \) is a state formula;
- if \( \phi \) and \( \psi \) are state formulas then so are \( \neg \phi \) and \( \phi \land \psi \);
- if \( \phi \) is a path formula then \( A \phi \) and \( E \phi \) are state formulas;
- if \( \phi \) is state formula and \( \omega \) is an organization then \( \text{MBEL}(\omega, \phi) \), \( \text{JGOAL}(\omega, \phi) \) and \( \text{JINTEND}(\omega, \phi) \) are state formulas; and
- if \( \omega_1 \) and \( \omega_2 \) are organizations then \( \text{subteam}(\omega_1, \omega_2) \) and \( \text{SUBTEAM}(\omega_1, \omega_2) \) is a state formula;
- if \( \phi \) is state formula and \( \omega_1 \) and \( \omega_2 \) are two organizations then \( \text{COMMAND}(\omega_1, \omega_2, \phi) \), \( \text{CONTROL}(\omega_1, \omega_2, \phi) \), and \( \text{COMMS}(\omega_1, \omega_2, \phi) \) are state formulas;
- if \( \phi \) and \( \psi \) are state formulas then \( X \phi \) and \( \phi \lor \psi \) are path formulas.

Path formulas of \( \text{BGIC}_{\text{CTL}} \) are restricted to be primitive linear-time temporal formulas, with no negations or disjunctions and no nesting of linear-time temporal operators.
Disjunctions, implications, and equivalences are defined in the classical way: $\phi \lor \psi$ is defined as $\neg(\neg\phi \land \neg\psi)$; $\phi \supset \psi$ is defined as $\neg(\phi \land \neg\psi)$; and $\phi \equiv \psi$ is defined as $\neg(\phi \land \neg\psi) \land \neg(\psi \land \neg\phi)$.

The operators MBEL, JGOAL, and JINTEND represent, respectively, the mutual beliefs, joint goals, and joint intentions of the relevant organization.

The proposition $\text{subteam}(\omega_i, \omega_j)$ represents the situation in which two teams are in a sub-team relationship. Its truth value is determined by a valuation function. The operator $\text{SUBTEAM}(\omega, \omega')$ represents a particular condition that involves the mutual beliefs of the organizations $\omega$ and $\omega'$ w.r.t the subteam proposition. This condition indicates that $\omega'$ is a sub-team of $\omega$.

The operators COMMAND, CONTROL, and COMMS represent, respectively, the command relationships, control relationships, and communication relationships between the two organizations.

8.3 Air Missions and Organization-Oriented System

Let us provide a complete analysis of the air mission scenario in the context of the organization-oriented approach. The air mission scenario example includes a number of organizations that adopt various organizational structures under different circumstances. In particular the relationships between the fighter controllers and the fighter pilots change over time.

We will first describe the organizations that are part of this example and the sub-team relationships between them. We will then proceed to describe the various organizational structures that these organizations can adopt. We will conclude with a description of the organizational plans that they employ in achieving their objectives.

8.3.1 Organizations

In the above example there are two primary organizations: Blue force and Orange force. We will focus here on the Blue force and the various sub-teams which are involved in its C3 structure.

Note that the team-oriented analysis of the air mission scenario has resulted in the specification of the teams involved in the scenario. The organization-oriented analysis results in similar entities but they are now organizations rather than teams. That is, there is also a specification of the organizational structure libraries that they should posses.

At the initial stage the Blue force has a sub-team relationship with a Sector Air Defence, AEW&C aircraft, AP-3C aircraft, and ADGE. The Sector Air Defence has a sub-team relationship with a SADC, and 2 pairs of hornets. We refer to these pairs as an Attack and a CAP respectively. The SADC is an agent and each hornet in the pair is also an agent. The AEW&C has a sub-team relationship with an MC and two FCs. In this scenario we view the AP-3C and ADGE as agents.
It is important to note that as described above the organizations change over time. In particular the two FCs on-board the AEW&C each join the Attack and CAP to form two new organizations. Organizations are joined by the MC to form the new AEW&C organization.

Figure 8.1: A specification of the FC organization family.

- AEW&C organization;
- Attack organization;
- CAP organization;
- a single SADC agent;
8.3. AIR MISSIONS AND ORGANIZATION-ORIENTED SYSTEMS

- four hornet agents;
- a single MC agent;
- two FC agents;
- a single AP-3C agent; and
- a single ADGE agent.

As noted earlier as part of this thesis we have developed a prototype development environment for building organization-oriented systems (see Chapter 12 for further details). We refer to this environment as the TOP system. In the TOP system each of the FCs is recognized as an instance of a type of organization. We refer to this as the organization family.

The particular organization family of an FC (see Figure 8.1) includes the specification of the knowledgebase for the organization (e.g., “fighter.db”), library of organizational plans available to the organization (e.g., “controller.mpl”), and library of organizational structures that an FC can adopt (e.g., “mission_example.org”). Note that this knowledge is stored in libraries that can be re-used in the specification of other organization families.

Please note that the prototype organization-oriented development environment is based on the dMARS™ agent-oriented development environment (see Chapter 12 for further details). Consequently some of the labels use the term “agent” instead of organization. Note that these labels do not refer to agents but to organizations.

8.3.2 Organizational Structures

Again we will focus on the organizational structures adopted by the Blue force. In particular we will focus on the structure adopted by the AEW&C and its sub-teams and the way they change over time.

As mentioned above at the initial stage the AEW&C includes three sub-teams: an MC and 2 FCs. In the organizational structure adopted by the AEW&C the MC is the commander and controller of the two FCs. On the formation of the two organizations, Attack and CAP, they adopt organizational structures.

Let us consider the possible organizational structures adopted by Attack. These organizational structures will depend on the joint goal it is attempting to achieve and the particular circumstances under which it is attempting to achieve these goals. In particular the choice of the organizational structure will depend on the level of autonomy that could be exercised by the Attack organization. This autonomy will depend on the information available to it (e.g., radar, visuals, etc.), the reliability of the radio communication, and other parameters.

If the Attack can operate relatively autonomously then the organizational structure adopted by Attack will include only command and communication relationships between the FC and the pair of hornets. Such type of organizational structure is referred to as
“tactical control C3 structure”. The specification of an organizational structure is done using a special editor for the specification of the tactical control structure.

The specification of the tactical control organizational structure (see Figure 8.2) includes the situation it is relevant for (i.e., invocation condition), a list of roles in the organization (e.g., “fc”), and a list of social relationships between the roles (e.g., communication relationship).

The specification of a social relationship between roles is done using a special form (see Figure 8.3). As an example the specification of the communication relationship in the tactical control organizational structure includes a specification of the senders, recipients, and the circumstances under which such communication should occur. Such circumstances include a particular state of the world, decisions that have been made, goals achieved, goals failed, etc.

If the Attack can not operate relatively autonomously then the organizational structure adopted by Attack will also include control relationship between the FC and the pair of
8.3. AIR MISSIONS AND ORGANIZATION-ORIENTED SYSTEMS

Figure 8.3: A specification of a communication relation.

hornets. This control will only be with respect to the choice of tactics (i.e., organizational plans) adopted by the attack. Such type of organizational structure is referred to as “close control C3 structure”. Again the specification of the organizational structure is done using the organizational structure editor.

The close control organizational structure includes a control relationship between the FC and the CAP (see Figure 8.4). This specification includes the specification of the controlling roles, controlled roles, and the type of control they exercise. That is, which means-end decisions do they control: choosing organizational structures, choosing organizational plans, choosing joint intentions, etc.

8.3.3 Organizational Plans and C3 Structures

We will now focus on the organizational plans available to the 2 hornets. These plans are stored in a plan library and are included in the specification of a fighter organization.
8.3. AIR MISSIONS AND ORGANIZATION-ORIENTED SYSTEMS

We will first consider two organizational plans that achieve a given joint goal required as part of the capabilities of a fighter organization. We will then proceed to describe two way in which the Command, Control, and Communication relationship are adhered to.

Organizational Plans

We will consider the plan (i.e., tactics) that achieves an intercept on a given target. In particular we will consider the pincer intercept. The pincer intercept involves the organization, with two sub-teams, forming into two distinct but cooperating elements in order to intercept an enemy aircraft. We identify two roles in the organizational structure: the lead and the wing roles (denoted by $\&$lead and $\&$wing respectively in Figure 8.5).

The tactics involves the lead role attacking the target from the left and the wing role attacking the target from the right (see Figure 8.5). The following briefly describes the stages of a pincer intercept:

- The joint goal to commence a pincer intercept is adopted by the organization (due
8.3. AIR MISSIONS AND ORGANIZATION-ORIENTED SYSTEMS

... to a command from a commanding organization or as a response to the situation) with a specified organizational structure.

- The lead and wing roles split to obtain lateral separation from the target.
- Both lead and wing roles perform a cutoff intercept on the target.
- Once the wing role completes the cutoff intercept it returns to a standard flight formation with the lead role.

Note that the executing organization is required to split into two separate components each flying in a different direction. It follows that this tactics can not be executed by an agent (denoted by the proposition singleton). This condition is checked in the context condition of the organizational plan. Also note that the pincer intercept is based on each of the sub-teams performing a cutoff intercept but from a different side.

Adhering to C3 Relationships

The TOP system implements a particular axiom system that establishes the relationships between the adopted organizational structure and the joint mental attitudes of the organizations that have adopted the structure. This relationship will be identified in detail our description of the operational semantics of organization-oriented systems (see Chapter 10).

The TOP system uses event based control mechanisms. Changes in the system state occur through the generation and handling of events. The relationships between the adopted organizational structure and the joint mental attitudes is implemented by the generation of special C3 events and default C3 event handlers that handle these events. The C3 default event handlers use physical communication as a mechanism of implementing the exchange of information and decisions.

In many cases physical communication may not be available. This is due to problems with the underlying communication infrastructure or a decision by the organization not to use communication due to costs, risks, or preferences. In the air mission domain we can find examples of both cases: (1) jamming of communications; and (2) a decision to observe radio silence because of operational reasons.

In such circumstances the adherence to the C3 structure has to be done by the sub-teams through the observation of the actions of the other sub-teams and the environment. This situation may allow for only limited coordination and high risk of confusion.

To allow for such type of coordination the designer of the organization-oriented system can override the default C3 event handlers. This is done by providing new C3 event handlers that respond to the special C3 events. As an example, consider a C3 event handler used by a controlled organization when there is radio silence.

The controlling organization will decide on the applicable tactic to be used by the controlled organization. The C3 event handler implements the observation of the decision made by the controlling organization. An example of such an observation could be that both organizations are flying in formation. On identification of a target the lead, which is...
8.3. AIR MISSIONS AND ORGANIZATION-ORIENTED SYSTEMS

Figure 8.5: An organizational plan for performing a pincer intercept.

in control of the wing, turns right. The wing may deduce from that they have commenced a cutoff intercept from the right.

Note that there is substantial room for confusion. Such confusion can come from the fact that the controlling organization may select a plan which is not considered applicable by the controlled organization. Alternatively there may be uncertainties as to the decision made by the controlling organization. In our example, if the lead would turn left the wing can deduce one of two things: (1) the lead has decided that the organization should perform a cutoff intercept from the left (in which case the wing is to follow the lead); or (2) the lead is decided that the organization should perform a pincer intercept and has just commenced its part of the cutoff from the left (in which case the wing is to commence a cutoff from the right).

These problems in implementing C3 structures with limited communication are very
8.4 Possible Worlds Semantics

In this section, we describe an extended possible-worlds semantics for social mental attitudes and joint plan execution. Again we adopt a possible worlds branching-time tree structure in which there are multiple possible worlds and each world has a branching-time tree structure. Multiple possible worlds model the organization’s lack of knowledge about the environment. Within each world, the branching future represents the choice of actions available to the organization and the chance, inherent in the environment, of the success or failure of these actions.

Let us first define an organization interpretation of the language to be an extension of a standard Kripke interpretation of possible worlds. The extension involves each possible world being a temporal structure. The edges are sets of organization-event pairs representing the events performed by the relevant organizations between these two time points.

**Definition 19**

Given a set $T$ of time points, a binary relation $\prec$ on $T$ which is total, transitive and backward-linear, a set of organization constants $T\mathcal{M}$, a set $\mathcal{E}$ of events, and an edge function $C$ that maps successor-restricted time point pairs to sets of organization-event pairs, a world or time tree, $w$ is a tuple $\langle T_w, \prec_w, T\mathcal{M}, \mathcal{E}, C_w \rangle$, where $T_w \subseteq T$ is a set of time points in the world $w$ and $\prec_w$ and $C_w$ are the same as $\prec$ and $C$ respectively, restricted to time points in $T_w$.

The definition of successor time points is identical to that given in Section 5.4. Again we restrict the edge function to allow for events to be performed only over successor time points.

**Definition 20**

An interpretation of an organization-oriented system, $M^{oos}$, is a tuple:

$$M^{oos} = \langle O, \mathcal{E}, T, \prec, C, W, \mathcal{MB}_i, \mathcal{JG}_i, \mathcal{JI}_i, \mathcal{CM}, \mathcal{CT}, \mathcal{CO}, U, \Phi \rangle$$

$O = \{\omega_1, \ldots, \omega_n\}$ is a set of organization constants, $\mathcal{E}$ is a set of events, $T$ is a set of time points. $\prec$ is a total, transitive, backwards-linear binary relation on $T$. $C$ is an edge function that maps successor-restricted time point pairs to sets of organization-event pairs. More formally, $C : \prec^* \rightarrow 2^{O \times \mathcal{E}}$. Intuitively, for any two time points for which the edge function $C$ is defined, it represents the events performed by the relevant organizations between those time points. $W$ is a set of worlds with respect to $T$, $\prec$, $\mathcal{E}$, and $C$. 

© Copyright Gil Tidhar, 1999 144
The accessibility relations, $MB_i$, $JG_i$, and $JI_i$ where $i \in O$ map a current state of the organization (i.e., a time point in a world) to the mutual-belief-, joint-goal-, and joint-intention-accessible worlds of the organization $i$, respectively. More formally, $MB_i \subseteq W \times T \times W$ and similarly for $JG_i$ and $JI_i$.

The accessibility relations, $CM$, $CT$, and $CM$ map for two organizations in $O$ a current state (i.e., a time point in a world) to the command-, control-, and communication-accessible worlds, respectively. More formally, $CM \subseteq O \times O \times W \times T \times W$ and similarly for $CT$ and $CO$.

$U$ is the universe of discourse and $\Phi$ is a mapping of first-order syntactic entities to relations over $U$ for any given world and time point.

Note again that we will at times refer to the above relations as function to the set $\{True, False\}$.

An organization $\omega$ has a mutual-belief $\phi$, denoted by $MBE_L(\omega, \phi)$, in a world $w$ at time point $t$ if and only if $\phi$ is true in all the mutual-belief-accessible worlds of the organization at time $t$ (and similarly for joint goals, and joint intentions).

Two organizations $\omega_1$ and $\omega_2$ have a command relationship with respect to $\phi$, denoted by $\text{COMMAND}(\omega_1, \omega_2, \phi)$, in a world $w$ at time point $t$ if and only if $\phi$ is true in all the command-accessible worlds at time $t$ (and similarly for control and communication relationships).

Satisfaction of formulas is denoted by $|=$. In our model satisfaction is with respect to the structure $M^{os}$, a world $w$, and a time point $t$. The expression $M^{os}, w_t |= \phi$ should be read as “structure $M^{os}$ in world $w$ and time point $t$ satisfies $\phi$”.

**Definition 21**

$M^{os}, w_t |= MBE_L(\omega, \phi)$ iff $\forall w'$ such that $MB(\omega, w, t, w')$, $M^{os}, w'_t |= \phi$;

$M^{os}, w_t |= JGOAL(\omega, \phi)$ iff $\forall w'$ such that $JG(\omega, w, t, w')$, $M^{os}, w'_t |= \phi$; and

$M^{os}, w_t |= JINTEND(\omega, \phi)$ iff $\forall w'$ such that $JI(\omega, w, t, w')$, $M^{os}, w'_t |= \phi$;

$M^{os}, w_t |= \text{COMMAND}(\omega, \omega', \psi)$ if and only if

$\forall w'$ such that $CM(\omega, \omega', w, t, w')$, we have $M^{os}, w'_t |= \psi$;

$M^{os}, w_t |= \text{CONTROL}(\omega, \omega', \psi)$ if and only if

$\forall w'$ such that $CT(\omega, \omega', w, t, w')$, we have $M^{os}, w'_t |= \psi$;

$M^{os}, w_t |= \text{COMMS}(\omega, \omega', \psi)$ if and only if

$\forall w'$ such that $CO(\omega, \omega', w, t, w')$, we have $M^{os}, w'_t |= \psi$.

The sub-team relation between two organizations is based on the mutual beliefs held by both organizations. We thus define the set $ST$ to represent the sub-team relation. That is, it defines for a given world and time point the organizations that are in a sub-team relationship.
We say that an organization $\omega_1$ has a sub-team $\omega_2$, in a world $w$ at time point $t$ if and only if the formula $\text{subteam}(\omega_1, \omega_2)$ is mutually believed by both organizations. More formally we define the $\text{ST}$ relation as follows:

**Definition 22**

$$\forall w \forall t \ (\omega_1, \omega_2, w, t) \in \text{ST} \text{ if and only if }$$

$$\forall v : (w, t, v) \in MB_w \text{ we have } M^{\text{eos}}, v_t \models \text{subteam}(\omega_1, \omega_2) \text{ and }$$

$$\forall v : (w, t, v) \in MB_{\omega_2} \text{ we have } M^{\text{eos}}, v_t \models \text{subteam}(\omega_1, \omega_2).$$

Note that $M^{\text{eos}}, w_t \models \text{subteam}(\omega_1, \omega_2)$ iff $\forall w'$ such that $MB_{\omega_1}(w, t, w')$, $M^{\text{eos}}, w'_t \models \text{subteam}(\omega_1, \omega_2)$. Based on the above definition we define the $\text{SUBTEAM}$ operator to represent the sub-team relation. That is, it defines the pairs of organizations that are in a sub-team relationship and is formally defined as follows:

**Definition 23**

$$M^{\text{eos}}, w_t \models \text{SUBTEAM}(\omega_1, \omega_2) \text{ if and only if } (\omega_1, \omega_2, w, t) \in \text{ST}.$$  

We restrict our discussion to models in which the sub-team relation is consistent across multiple worlds but can change over time. More formally:

**Definition 24**

If $M^{\text{eos}}, w_t \models \text{SUBTEAM}(\omega_1, \omega_2)$ then $\forall v M^{\text{eos}}, v_t \models \text{SUBTEAM}(\omega_1, \omega_2)$

We also extend the K-axioms and generalization rules to cover the social mental attitudes. More formally we have:

**CM-K** $\text{COMMAND}(\omega_1, \omega_2, \phi) \land \text{COMMAND}(\omega_1, \omega_2, \phi \supset \psi) \supset \text{COMMAND}(\omega_1, \omega_2, \psi)$

**CT-K** $\text{CONTROL}(\omega_1, \omega_2, \phi) \land \text{CONTROL}(\omega_1, \omega_2, \phi \supset \psi) \supset \text{CONTROL}(\omega_1, \omega_2, \psi)$

**CO-K** $\text{COMMS}(\omega_1, \omega_2, \phi) \land \text{COMMS}(\omega_1, \omega_2, \phi \supset \psi) \supset \text{COMMS}(\omega_1, \omega_2, \psi)$

**CM-Gen** If $\vdash \phi$ then $\vdash \text{COMMAND}(\omega_1, \omega_2, \phi)$

**CT-Gen** If $\vdash \phi$ then $\vdash \text{CONTROL}(\omega_1, \omega_2, \phi)$

**CO-Gen** If $\vdash \phi$ then $\vdash \text{COMMS}(\omega_1, \omega_2, \phi)$

Following the nomenclature adopted in Chapter 6 the above axiom system is denoted $\text{C3BGI}_{\text{CTL}}^{\text{eos}}$. 

© Copyright Gil Tidhar, 1999 146
8.5 Soundness and Completeness

The C3BGI\textsuperscript{C3} CTL axiom system and model described above are an extension to the team-oriented model described in Chapter 6. Here we will modify the proof of soundness and completeness for a team-oriented model to take into account the new concepts introduced.

The primary difference between the two models is the introduction of the command, control, and communication relationships between organizations. We first define the consistency rules for the C3 component.

**Basic C3-Consistency Rules:**

1. **CM0** if \( \text{COMMAND}(\omega_1, \omega_2, \phi) \in \Phi(w_t) \) and \((\omega_1, \omega_2, w, t, v) \in \mathcal{CM})\ then \phi \in \Phi(v_t);
2. **CM1** if \( \neg \text{COMMAND}(\omega_1, \omega_2, \phi) \in \Phi(w_t) \) then \( \exists v \) such that \((\omega_1, \omega_2, w, t, v) \in \mathcal{CM})\ and \neg \phi \in \Phi(v_t);
3. **CT0** if \( \text{CONTROL}(\omega_1, \omega_2, \phi) \in \Phi(w_t) \) and \((\omega_1, \omega_2, w, t, v) \in \mathcal{CT})\ then \phi \in \Phi(v_t);
4. **CT1** if \( \neg \text{CONTROL}(\omega_1, \omega_2, \phi) \in \Phi(w_t) \) then \( \exists v \) such that \((\omega_1, \omega_2, w, t, v) \in \mathcal{CT})\ and \neg \phi \in \Phi(v_t);
5. **CO0** if \( \text{COMMS}(\omega_1, \omega_2, \phi) \in \Phi(w_t) \) and \((\omega_1, \omega_2, w, t, v) \in \mathcal{CO})\ then \phi \in \Phi(v_t);
6. **CO1** if \( \neg \text{COMMS}(\omega_1, \omega_2, \phi) \in \Phi(w_t) \) then \( \exists v \) such that \((\omega_1, \omega_2, w, t, v) \in \mathcal{CO})\ and \neg \phi \in \Phi(v_t);

We now define the following quotient construction for \( M^{\text{Mtos}} \):

**Definition 25**

The quotient construction for \( M^{\text{Mtos}} \) is an extension to the quotient construction for \( M^{\text{Mtos}} \) with the addition of \( \mathcal{CM}^q \), \( \mathcal{CT}^q \), and \( \mathcal{CO}^q \). Where \( \mathcal{CM}^q = \{ (\omega_1, \omega_2, [w_t], [v_t]) : \text{COMMAND}^-(\omega_1, \omega_2, [w_t]) \subseteq [v_t] \} \). Where \( \text{COMMAND}^-(\omega_1, \omega_2, X) = \{ \phi : \text{COMMAND}(\omega_1, \omega_2, \phi) \in X \} \). Similarly we define \( \mathcal{CT}^q \), \( \mathcal{CO}^q \), \( \text{CONTROL}^-(\omega_1, \omega_2, X) \), and \( \text{COMMS}^-(\omega_1, \omega_2, X) \).

A C3BGI\textsuperscript{C3} CTL-tableau is a BGI\textsuperscript{Mtos} CTL-tableau modified to include the C3-consistency rules. More formally:

**Definition 26**

Given a formula \( \phi_0 \), a C3BGI\textsuperscript{C3} CTL-tableau for \( \phi_0 \) is a tuple \( \langle O, E, T, \prec, C, W, MB_i, JG_i, JI_i, \mathcal{CM}, \mathcal{CT}, \mathcal{CO}, U, \Phi \rangle \) (with \( \phi_0 \in \Phi(w_t) \) for some \( w \in W \) and some \( t \in T_w \in T \)) which meets the following conditions:

1. the conditions of the BGI\textsuperscript{Mtos} CTL-tableau are satisfied;
2. the basic C3-consistency rules (CM0), (CM1), (CT0), (CT1), (CO0), and (CO1) are satisfied; and

Similarly, one can define a pseudo-C3BGI\textsuperscript{\textit{oos}}\_CTL-tableau. Let us now prove the small model theorem for C3BGI\textsuperscript{\textit{oos}}\_CTL.

**Theorem 11** Let $\phi_0$ be a BGI\_CTL formula of length $n$, then the following equivalences hold:

1. $\phi_0$ is C3BGI\textsuperscript{\textit{oos}}\_CTL-satisfiable.;
2. $\phi_0$ has a model $M\textsuperscript{\textit{oos}}$ with a finite branching in each world bounded by $O(n)$;
3. $\phi_0$ has a finite pseudo-C3BGI\textsuperscript{\textit{oos}}\_CTL-tableau of size $\leq \exp(n)$;
4. $\phi_0$ has a finite model $M\textsuperscript{\textit{oos}}$ of size $\leq \exp(n)$.

**Proof:**

For the CTL and BGI components the proof is identical to the proof of Theorem 5. For the C3 component we need to make the appropriate changes to the proof that (2)$\rightarrow$(3) and show that the models in $M\textsuperscript{\textit{oos}}$ satisfy the consistency rules (CM0), (CM1), (CT0), (CT1), (CO0), and (CO1).

Let $M\textsuperscript{\textit{oos}}$ be a class of structures with $M \in M\textsuperscript{\textit{oos}}$ such that $M, w_t \models \phi_0$. We show that $M^q = M/\equiv_{\text{rel}(\phi_0)}$ is a pseudo-BGI\textsuperscript{\textit{oos}}\_CTL-tableau. We show that the quotient construction satisfies (CM0). The other conditions can be proved in the same way.

Let $\text{COMMAND}(\omega_1, \omega_2, \phi) \in \Phi(w_t)$ and $\text{CM}(\omega_1, \omega_2, w_t, v)$. From the quotient construction we have $M^q, [w_t] \models \text{COMMAND}(\omega_1, \omega_2, \phi)$ and $M^q([w_t], [v])$. From the definition of command we have $M^q, [v] \models \phi$ and as a result $\phi \in \Phi(v_t)$.

From the construction the size of the model can be shown to be $l * N^2$ where $l$ is the number of eventualities and $N$ is the number of nodes in the model. The number of nodes $N \leq 2^n$ where $n$ is the length of $\phi_0$. We thus get that the size of the model $\leq \exp(n)$.

Note that this number is independent of the number of agents but rather depends on the complexity of the belief, goal, and intention accessibility relations.

Let us now describe the algorithm for constructing a pseudo-BGI\textsuperscript{\textit{oos}}\_CTL-tableau. Again the algorithm is essentially the same algorithm described in Chapter 6 with some changes made to take into account the the introduction of the command, control, and communication relations. The changes are made to Step 2(d) as follows:

2(d) Expanding elementary BGI formula: If node $w_t$ is a leaf of the tree, $\Phi(w_t)$ is not blatantly inconsistent, and $\Phi(w_t)$ is fully expanded propositional tableau, then:
8.5. SOUNDNESS AND COMPLETENESS

Figure 8.6: An example of a pseudo-BGI<sub>CS</sub>-tableau.

1. if \( \Phi(w_t) \) contains \( \neg \text{MBEL}(\omega_{i_1}, \phi_1) \ldots \neg \text{MBEL}(\omega_{i_m}, \phi_m) \) then create \( m \mathcal{M}B_{\omega_{i_j}} \)-successors of node \( w_{t_i} \), labelled with \( \text{MBEL}^-(\omega_{i_j}, \Phi(w_t)) \cup \text{SUBTEAM}^-(\omega_{i_j}, \Phi(w_t)) \cup \{\neg \phi_j\} \), where \( 1 \leq j \leq m \);

2. if \( \Phi(w_t) \) contains \( \neg \text{JGOAL}(\omega_{i_1}, \phi_1) \ldots \neg \text{JGOAL}(\omega_{i_m}, \phi_m) \) then create \( m \mathcal{J}G_{\omega_{i_j}} \)-successors of node \( w_{t_i} \), labelled with \( \text{JGOAL}^-(\omega_{i_j}, \Phi(w_t)) \cup \{\neg \phi_j\} \), where \( 1 \leq j \leq m \);

3. if \( \Phi(w_t) \) contains \( \neg \text{JINTEND}(\omega_{i_1}, \phi_1) \ldots \neg \text{JINTEND}(\omega_{i_m}, \phi_m) \) then create \( m \mathcal{J}T_{\omega_{i_j}} \)-successors of node \( w_{t_i} \), labelled with \( \text{JINTEND}^-(\omega_{i_j}, \Phi(w_t)) \cup \{\neg \phi_j\} \), where \( 1 \leq j \leq m \);

4. if \( \Phi(w_t) \) contains \( \neg \text{SUBTEAM}(\omega_{i_1}, \omega'_{i_1}) \ldots \neg \text{SUBTEAM}(\omega_{i_m}, \omega'_{i_m}) \) then create \( m \mathcal{M}B_{\omega_{i_j}} \)-successors of node \( w_{t_i} \), labelled with \( \text{SUBTEAM}^-(\omega_{i_j}, \Phi(w_t)) \cup \text{MBEL}^-(\omega_{i_j}, \Phi(w_t)) \cup \{\neg \text{subteam}(\omega_{i_j}, \omega'_{i_j})\} \), where \( 1 \leq j \leq m \) and also create \( m \mathcal{M}B_{\omega_{i_j}} \)-successors of node \( w_{t_i} \), labelled with \( \text{SUBTEAM}^-(\omega_{i_j}, \Phi(w_t)) \cup \text{MBEL}^-(\omega_{i_j}, \Phi(w_t)) \cup \{\neg \text{subteam}(\omega_{i_j}, \omega'_{i_j})\} \), where \( 1 \leq j \leq m \);

5. if \( \Phi(w_t) \) contains \( \neg \text{COMMAND}(\omega_{i_1}, \omega'_{i_1}, \phi_1) \ldots \neg \text{COMMAND}(\omega_{i_m}, \omega'_{i_m}, \phi_m) \) then create \( m \mathcal{C}M \)-successors of node \( w_{t_i} \), labelled with \( \text{COMMAND}^-(\omega_{i_j}, \omega'_{i_j}, \Phi(w_t)) \cup \{\neg \phi_j\} \), where \( 1 \leq j \leq m \);

6. if \( \Phi(w_t) \) contains \( \neg \text{CONTROL}(\omega_{i_1}, \omega'_{i_1}, \phi_1) \ldots \neg \text{CONTROL}(\omega_{i_m}, \omega'_{i_m}, \phi_m) \) then create \( m \mathcal{C}T \)-successors of node \( w_{t_i} \), labelled with \( \text{CONTROL}^-(\omega_{i_j}, \omega'_{i_j}, \Phi(w_t)) \cup \{\neg \phi_j\} \), where \( 1 \leq j \leq m \);

7. if \( \Phi(w_t) \) contains \( \neg \text{COMMS}(\omega_{i_1}, \omega'_{i_1}, \phi_1) \ldots \neg \text{COMMS}(\omega_{i_m}, \omega'_{i_m}, \phi_m) \) then create \( m \mathcal{C}O \)-successors of node \( w_{t_i} \), labelled with \( \text{COMMS}^-(\omega_{i_j}, \omega'_{i_j}, \Phi(w_t)) \cup \{\neg \phi_j\} \), where
1 ≤ j ≤ m;

Consider the simple tableau illustrated in Figure 8.6. Note that in this case the command relationship between organizations a and b has no impact on nodes M0 or M1. That is the joint goals (or the negation of such) held by the two teams is independent of the command relationships.

Given a formula \( \phi \) and a structure \( M_{\text{oos}} = \langle T, \mathcal{E}, \prec, \mathcal{C}, W, \mathcal{S}, MB, \mathcal{G}, \mathcal{J}, \mathcal{C}, \mathcal{T}, \mathcal{O}, U, \Phi \rangle \), we define \(|W|\) to be the number of worlds in \( W \), \(|BGI_{\text{oos}}|\) to be the total number of tuples in \( MB_{\omega}, \mathcal{G}_{\omega}, \mathcal{J}, \mathcal{C}, \mathcal{T}, \mathcal{O} \), and \(|ST|\) to be number of sub-team relations, and \(|\phi|\) to be the length of \( \phi \).

The only difference in complexity between the algorithm described above and the algorithm presented in Chapter 6 is the addition of the consideration of the command, control, and communication relations. We thus need to add to the complexity the number of such relations considered. That is, \(|CM|\), \(|CT|\), and \(|CO|\).

Given the above construction we get that: for a given formula \( \phi \) and structure \( M_{\text{oos}} = \langle T, \mathcal{E}, \prec, \mathcal{C}, W, \mathcal{S}, MB, \mathcal{G}, \mathcal{J}, \mathcal{C}, \mathcal{T}, \mathcal{O}, U, \Phi \rangle \), there is an algorithm for checking if \( \phi \) is satisfied in \( M_{\text{oos}} \) that runs in time \( O((|W| + |BGI_{\text{oos}}| + |ST|) \times |\phi|) \).

To show that BGI\( _{\text{CTL}} \)-system is sound and complete we need to show that a formula \( \phi_0 \) is BGI\( _{\text{CTL}} \)-provable if and only if \( \phi_0 \) is marked ‘satisfiable’ in a pseudo-BGI\( _{\text{oos}} \)-tableau. To show this we will need to prove the following two lemmas that state that in the pseudo-BGI\( _{\text{oos}} \)-tableau if a label of node is inconsistent then the label of its \( \prec \) and \( \mathcal{C} \mathcal{M} \) predecessors are also inconsistent. These two lemmas are similar to Lemma 1 and Lemma 2 presented by Rao and Georgeff [90] (pp. 41). Note that we have introduced the K-axioms and generalization rules for the social mental attitudes. The proof of Lemma 5 is identical to the proof given by Rao and Georgeff [90] and will not be provided here. The proof of Lemma 6 is a slight variation of the proof by Rao and Georgeff and is provided below.

Lemma 5 Let \( \Phi_m \) and \( \Phi_n \) be conjunctions of propositions in \( m \) and \( n \) respectively and \((n,m) \in \prec \) as constructed in the pseudo-BGI\( _{\text{CTL}} \)-tableau then if \( \Phi_m \) is inconsistent then \( \Phi_n \) is also inconsistent.

Lemma 6 Let \( \Phi_m \) and \( \Phi_n \) be conjunctions of propositions in \( m \) and \( n \) respectively and \((\omega_1,\omega_2,n,m) \in \mathcal{C} \mathcal{M} \) as constructed in the pseudo-BGI\( _{\text{CTL}} \)-tableau then if \( \Phi_m \) is inconsistent then \( \Phi_n \) is also inconsistent.

Proof: (Lemma 6)
Suppose \((\omega_1,\omega_2,n,m) \in \mathcal{C} \mathcal{M} \). By construction, for some formula \( \neg \text{COMMAND}(\omega_1,\omega_2,\psi_k) \) in node \( n \) such that \( \text{COMMAND}^{-}(\omega_1,\omega_2,n) = \{\phi_1, \ldots, \phi_z\} \), we have, \( \phi_1, \ldots, \phi_z, \neg \psi_k \) in node \( m \).

1. \( \vdash \phi_1 \rightarrow (\phi_2 \rightarrow (\ldots (\phi_z \rightarrow \psi_k \ldots)) \) {Assumption that \( \Phi_m \) is inconsistent and propositional reasoning}
2. ⊢ \text{COMMAND}(\omega_1, \omega_2, \phi_1 \rightarrow (\phi_2 \rightarrow (\ldots (\phi_z \rightarrow \psi_k) \ldots))) \{\text{Generalization Rule CM-Gen}\}

3. ⊢ \text{COMMAND}(\omega_1, \omega_2, \phi_1 \rightarrow (\phi_2 \rightarrow (\ldots (\phi_z \rightarrow \psi_k) \ldots))) \rightarrow (\text{COMMAND}(\omega_1, \omega_2, \phi_1) \\
\rightarrow (\text{COMMAND}(\omega_1, \omega_2, \phi_2) \rightarrow (\ldots (\text{COMMAND}(\omega_1, \omega_2, \phi_2) \rightarrow \text{COMMAND}(\omega_1, \omega_2, \psi_k)) \\
\ldots))) \{\text{From Axiom CM-K}\}

4. ⊢ \text{COMMAND}(\omega_1, \omega_2, \phi_1) \rightarrow (\text{COMMAND}(\omega_1, \omega_2, \phi_2) \rightarrow (\ldots (\text{COMMAND}(\omega_1, \omega_2, \phi_2) \rightarrow \text{COMMAND}(\omega_1, \omega_2, \psi_k))) \\
\ldots))) \{\text{Propositional Reasoning}\}

5. ⊢ \neg(\text{COMMAND}(\omega_1, \omega_2, \phi_1) \land \text{COMMAND}(\omega_1, \omega_2, \phi_2) \land \ldots \text{COMMAND}(\omega_1, \omega_2, \phi_z) \land \neg \text{COMMAND}(\omega_1, \omega_2, \psi_k)) \{\text{Propositional Reasoning}\}

\[\blacksquare\]

**Theorem 12** The BGI\textsuperscript{os}\textsubscript{CTL}-system is a sound and complete axiomatization with respect to \(M\textsuperscript{os}\).

**Proof:**

The proof for the joint mental attitudes is essentially identical to the proof provided in Chapter 6 for Theorem 6. The only modification is with respect to Case 3, i.e., node \(n\) is an internal node of the tree and is a fully expanded propositional CTL tableau. Here we need to consider the case of the C3 accessibility relations, CM, CT, and CO. We thus need to consider the C3, CT, and CO-successors.

Note that the proof of soundness and completeness is actually based on the pseudo-tableau construction and the inconsistency lemmas. Also note that we have changed the pseudo-tableau construction algorithm in a way that corresponds to the changes above. It follows that the proof for soundness and completeness for the social mental attitudes is identical.

\[\blacksquare\]

### 8.6 The Spectrum of Social and Joint Mental Attitudes

In Chapter 6 we introduced the notion of a team-oriented system and joint mental attitudes. In Chapter 7 we described a variety of team-oriented systems that are based on the different relationships between the joint mental attitudes and team structure. In the previous sections we introduced the notion of an organization-oriented system and social mental attitudes.

Both social and joint mental attitudes are modelled as separate accessibility relations over a given set of worlds. We view the relationship between these attitudes as containment conditions between the different relations. One can explore the full spectrum of
organization-oriented systems by looking at any combination of containment conditions. This is similar to the approach of Rao and Georgeff [90] in exploring the spectrum of agent-oriented systems.

It is clear that from the full spectrum of organization-oriented systems there are some that will have “interesting” characteristics and others that will not. In this work the criteria used to evaluate whether a particular organization-oriented system is “interesting” is as follows:

1. The organization-oriented system is a model of a particular type of organization recognized in Organization Theory; or
2. The organization-oriented system has characteristics that are desirable for the purpose of developing a particular decentralized computer system.

In the following sections we explore parts of the spectrum of organization-oriented systems. We divide the discussion to: (1) the exploration of the relationships between the different social mental attitudes; (2) the exploration of the relationships between the social mental attitudes and the joint mental attitudes; and (3) the exploration of the relationships between the social mental attitudes and the structure of the organization.

Recall that the semantics of the social mental attitudes between two organizations are defined as accessibility relations between the possible worlds. These relations can be viewed in a standard way as sets of worlds. The possible relationships between the different social mental attitudes can be described as the possible set relations between the sets of worlds.

These set relations include: equivalence of sets, inclusion of one set in another set, empty and non empty intersections of sets. We only explore some of the possible set relationships between the different social mental attitudes, joint mental attitudes, and team structures that may exist between two organizations. Other set relations can be described in a similar way.

We denote by $S$ one of the sets $CM$, $CT$, or $CO$. We denote by $S(\omega, \omega')$ the set of possible worlds for the relationship between the two organizations $\omega$ and $\omega'$. We denote by $S$ one of the operators $COMMAND$, $CONTROL$, or $COMMS$. We denote by $S(\omega, \omega')$ the social mental attitude between the organizations $\omega$ and $\omega'$. We use the above notation to define classes of relations between different social mental attitudes.

We provide the intuition and examples for each such set relation described. In Chapter 9 we provide an elaborate discussion and soundness and completeness proofs for a number of “interesting” organization-oriented systems.

**Example 2**

Let us consider again the simple organizational structure and categorize it using the possible relations between the social mental attitudes, joint mental attitudes, and team structure. We will denote the simple organization as $SO$, the operating core as $OC$, and the members of the operating core as $w_i$ where $1 \leq i \leq n$. We denote one of the objectives of the simple organization as $\phi$. We can define the team structure and different social
relationships between the different organizations as a the following set of propositions:

1. $\text{SUBTEAM}(SO, CEO)$
2. $\text{SUBTEAM}(SO, OC)$
3. $\text{SUBTEAM}(OC, w_i), \ 1 \leq i \leq n$
4. $\text{COMMAND}(CEO, SO, \phi)$
5. $\text{CONTROL}(CEO, SO, \phi)$
6. $\text{COMMAND}(CEO, OC, \phi)$
7. $\text{CONTROL}(CEO, OC, \phi)$
8. $\text{COMMAND}(CEO, w_i, \phi), \ 1 \leq i \leq n$
9. $\text{CONTROL}(CEO, w_i, \phi), \ 1 \leq i \leq n$
10. $\text{COMMS}(CEO, SO, \phi)$
11. $\text{COMMS}(SO, CEO, \phi)$
12. $\text{COMMS}(CEO, OC, \phi)$
13. $\text{COMMS}(OC, CEO, \phi)$
14. $\text{COMMS}(CEO, w_i, \phi), \ 1 \leq i \leq n$
15. $\text{COMMS}(w_i, CEO, \phi), \ 1 \leq i \leq n$
16. $\text{COMMAND}(CEO, SO, \phi) \land \text{JGOAL}(CEO, \phi) \supset \text{JGOAL}(SO, \phi)$
17. $\text{CONTROL}(CEO, SO, \phi) \land \text{JINTEND}(CEO, \phi) \supset \text{JINTEND}(SO, \phi)$
18. $\text{COMMAND}(CEO, OC, \phi) \land \text{JGOAL}(CEO, \phi) \supset \text{JGOAL}(OC, \phi)$
19. $\text{CONTROL}(CEO, OC, \phi) \land \text{JINTEND}(CEO, \phi) \supset \text{JINTEND}(OC, \phi)$
20. $\text{COMMAND}(CEO, w_i, \phi) \land \text{JGOAL}(CEO, \phi) \supset \text{JGOAL}(w_i, \phi), \ 1 \leq i \leq n$
21. $\text{CONTROL}(CEO, w_i, \phi) \land \text{JINTEND}(CEO, \phi) \supset \text{JINTEND}(w_i, \phi), \ 1 \leq i \leq n$

The above social mental attitudes can be expressed more simply using a number of simple axiom schema that determine the relationship between command, control, communication, and team structure. We maintain propositions (1)–(4), (6), and (8) and add to the them the following axioms.
1. Stating that the authority and control are held by the same member of the organization is represented by equating the command and control relationships. That is, \( \text{COMMAND}(\alpha, \omega, \phi) \equiv \text{CONTROL}(\alpha, \omega, \phi) \).

2. Stating that the authority and control are centralized is represented as the transitivity of command and control through the team structure. That is, a commander (or controller) of an organization is also the commander (or controller) of any of the sub-teams of that organization: \( \text{COMMAND}(\alpha, \omega, \phi) \land \text{SUBTEAM}(\omega, \omega') \supset \text{COMMAND}(\alpha, \omega', \phi) \). We have a similar axiom for control: \( \text{CONTROL}(\alpha, \omega, \phi) \land \text{SUBTEAM}(\omega, \omega') \supset \text{CONTROL}(\alpha, \omega', \phi) \). Note that given that \( w_i \) is a sub-team of \( OC \) we will get that: \( \text{COMMAND}(OC, w_i, \phi) \) as required.

3. Stating that the communication links are directly between the CEO and the members of the operating core is represented as the command relationship implying communication relationships between the commanding and commanded organizations. That is, \( \text{COMMAND}(\alpha, \omega, \phi) \supset \text{COMMS}(\alpha, \omega, \phi) \land \text{COMMS}(\omega, \alpha, \phi) \).

4. Stating that the authority of the CEO is reflected in the ability to delegate goals and control the way these goals are achieved is represented as the command relationship and an adopted goal by the commanding organization implying the corresponding adopted goal by the commanded organization. That is, \( \text{COMMAND}(\alpha, \omega, \phi) \land \text{JGOAL}(\alpha, \phi) \supset \text{JGOAL}(\omega, \phi) \). A similar axiom exists for control relationships, \( \text{CONTROL}(\alpha, \omega, \phi) \land \text{JINTEND}(\alpha, \phi) \supset \text{JINTEND}(\omega, \phi) \).

8.7 Interactions Between Different Social Mental Attitudes

Social mental attitudes are represented as a relationship between two organizations and a state formula. The relationship between the two organizations is not necessarily symmetric. The interactions between the different social mental attitudes are with respect to all social mental attitudes that exist between two given organizations, that is, the social mental attitudes from one organization to a second organization as well as the possible attitudes from the second organization to the first. Furthermore, these interactions include the social mental attitudes that exist between a third organization and any of the two organizations.

We will start by only considering the interactions between the different social mental attitudes that exist between one organization and another. We will ignore any social mental attitudes that exist between any other organization and either of the two organizations. We refer to this situation as uni-directional social mental attitudes. We will then proceed to also consider the additional interactions between the second organization and the first organization. We refer to this situation as bi-directional social mental attitudes.
In Table 8.1 we use \( \alpha_i \), where \( 1 \leq i \leq 4 \), as place holders in which any of \( \omega \) or \( \omega' \) could be placed to form different types of conditions and axioms for uni-directional, bi-directional, and multi-directional social mental attitudes.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Axiom</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_1(\alpha_1, \alpha_2) = S_2(\alpha_3, \alpha_4) )</td>
<td>( S_2(\alpha_3, \alpha_4, \phi) \equiv S_1(\alpha_1, \alpha_2, \phi) )</td>
</tr>
<tr>
<td><strong>Description:</strong> The situation in which two social mental attitudes always exist together. An example of such a situation is that the commander and controller is the same organization as is the case in the simple organizational structure described above: ( \text{COMMAND}(\omega, \omega', \phi) \equiv \text{CONTROL}(\omega, \omega', \phi) ).</td>
<td></td>
</tr>
<tr>
<td>( S_1(\alpha_1, \alpha_2) \subseteq S_2(\alpha_3, \alpha_4) )</td>
<td>( S_2(\alpha_3, \alpha_4, \phi) \supset S_1(\alpha_1, \alpha_2, \phi) )</td>
</tr>
<tr>
<td><strong>Description:</strong> The situation in which if one social mental attitude exists then the other social mental attitude will not exist. An example of such a situation is that if one organization commands another organization then the second organization can not command the first organization: ( \text{COMMAND}(\omega, \omega', \phi) \supset \neg \text{COMMAND}(\omega, \omega', \phi) ).</td>
<td></td>
</tr>
<tr>
<td>( S_1(\alpha_1, \alpha_2) \cap S_2(\alpha_3, \alpha_4) = \emptyset )</td>
<td>( S_2(\alpha_3, \alpha_4, \phi) \supset \neg S_1(\alpha_1, \alpha_2, \phi) )</td>
</tr>
<tr>
<td><strong>Description:</strong> The situation in which there is consistency between the authority or responsibility of an organization in its social mental attitudes with another organization. If one social mental attitude towards a particular state formula exist then the other social mental attitude towards the negation of that state formula will not exist. An example of such a situation is that the commander of an organization with respect to ( \phi ) does not want to be communicated the negation of ( \phi ): ( \text{COMMAND}(\omega, \omega', \phi) \supset \neg \text{COMMS}(\omega, \omega, \neg \phi) ).</td>
<td></td>
</tr>
</tbody>
</table>

Table 8.1: Uni and Bi-Directional Organization-Oriented Modal Systems (a)

In Table 8.1 we only expressed possible relationships between two social mental attitudes. Given that there are three social mental attitudes there can also be relationships that involve three social mental attitudes. We explore these relationships in Table 8.2 and use \( \alpha_i \), where \( 1 \leq i \leq 6 \), as place holders in which any of \( \omega \) or \( \omega' \) could be placed to form different types of conditions and axioms for uni-directional, bi-directional, and multi-directional social mental attitudes.

### 8.8 Interaction Between Social Mental Attitudes and Joint Mental Attitudes

In this section we consider the interaction between the social mental attitudes that exist between organizations and the joint mental attitudes held by these organizations. We denote by \( J_\omega \) one of the sets \( JG_\omega \), \( JI_\omega \), or \( MB_\omega \). We denote by \( J(\omega) \) one of the operators \( \text{JGOAL}(\omega) \), \( \text{JINTEND}(\omega) \), or \( \text{MBEL}(\omega) \).

We use the above notation to define classes of relations between different social and joint mental attitudes. In Tables 8.3 and 8.4 we use \( \alpha_i \), where \( 1 \leq i \leq 5 \), as place holders
8.8. INTERACTION BETWEEN SOCIAL MENTAL ATTITUDES AND JOINT MENTAL ATTITUDES

CHAPTER 8. ORGANIZATION-ORIENTED SYSTEMS

<table>
<thead>
<tr>
<th>Condition</th>
<th>Axiom</th>
</tr>
</thead>
<tbody>
<tr>
<td>${S_1(\alpha_1, \alpha_2) \cup S_2(\alpha_3, \alpha_4)} \subseteq S_3(\alpha_5, \alpha_6)$</td>
<td>$S_3(\alpha_5, \alpha_6, \phi) \supset S_1(\alpha_1, \alpha_2, \phi) \cup S_2(\alpha_3, \alpha_4, \phi)$</td>
</tr>
<tr>
<td><strong>Description:</strong> The situation in which if one social mental attitude exists then so will two other social mental attitudes but not vice versa. An example of such a situation is that control requires communication of both organizations towards each other. These relationships correspond to the performance information flow as described above: $\text{CONTROL}(\omega, \omega', \phi) \supset \text{COMMS}(\omega, \omega', \phi) \land \text{COMMS}(\omega', \omega, \phi)$.</td>
<td></td>
</tr>
</tbody>
</table>

| $S_3(\alpha_5, \alpha_6) \subseteq \{S_1(\alpha_1, \alpha_2) \cup S_2(\alpha_3, \alpha_4)\}$ | $S_1(\alpha_1, \alpha_2, \phi) \land S_2(\alpha_3, \alpha_4, \phi) \supset S_3(\alpha_5, \alpha_6, \phi)$ |
| **Description:** The situation in which if two social mental attitudes exist then so will a third social mental attitude. An example of such a situation is the “chain of command” and the authority that a commander of a commander has over an organization. That is, if an organization commands a second organization and that organization commands a third organization then the first organization also commands the third organization: $\text{COMMAND}(\omega, \omega', \phi) \land \text{COMMAND}(\omega', \omega'', \phi) \supset \text{COMMAND}(\omega, \omega'', \phi)$. |

| $S_3(\alpha_5, \alpha_6) \cap \{S_1(\alpha_1, \alpha_2) \cup S_2(\alpha_3, \alpha_4)\} = \emptyset$ | $S_1(\alpha_1, \alpha_2, \phi) \land S_2(\alpha_3, \alpha_4, \phi) \supset \neg S_3(\alpha_5, \alpha_6, \phi)$ |
| **Description:** The situation in which the separation of authority or responsibility. If two social mental attitudes exist then the other social mental attitude will not exist. An example of such a situation is the “arrogant boss”. That is, the commander and control of an organization will not communicate with the organization it commands and controls: $\text{COMMAND}(\omega, \omega', \phi) \land \text{CONTROL}(\omega', \omega, \phi) \supset \neg \text{COMMS}(\omega, \omega', \phi)$. Another example of such a situation is the “silent employee”. That is, the commanded organization does not communicate to the commander that communicates with it: $\text{COMMAND}(\omega, \omega', \phi) \land \text{COMMS}(\omega, \omega', \phi) \supset \neg \text{COMMS}(\omega, \omega')$. |

Table 8.2: Uni and Bi-Directional Organization-Oriented Modal Systems (b)

in which any of $\omega$ or $\omega'$ could be placed to form different types of conditions and axioms for uni-directional and bi-directional social mental attitudes.

In this section we define a number of axiom systems that formalize different notions of social rationality. We start with the basic approach formalizing the notion of command, control, and communications described in the introduction to this chapter.

8.8.1 Basic Organization-Oriented System

The basic approach involves the following relationship between the social mental attitudes and the joint mental attitudes: (1) a commanding organization can cause the adoption of a joint goal by the commanded organization; (2) a controlling organization can cause the adoption of a joint intention by a controlled organization; and (3) a communicating organization causes the adoption of a mutual belief by the recipient organization. More formally we have:

(CM-BA) $\text{COMMAND}(\omega, \omega', \phi) \land \text{JGOAL}(\omega, \phi) \supset \text{JGOAL}(\omega', \phi)$;

(CT-BA) $\text{CONTROL}(\omega, \omega', \phi) \land \text{JINTEND}(\omega, \phi) \supset \text{JINTEND}(\omega', \phi)$;

(CO-BA) $\text{COMMS}(\omega, \omega', \phi) \land \text{MBEL}(\omega, \phi) \supset \text{MBEL}(\omega', \phi)$.

We denote an organization-oriented system in which the above axioms hold as the $\text{BGI}^{\text{CM-BA} \land \text{CT-BA} \land \text{CO-BA}}$ system. Note that if we consider a single organization and its social mental attitudes with itself then the command, control, and communication relationships are reflexive with respect to every organization. It follows that for every organization it can
### 8.8. Interaction Between Social Mental Attitudes and Joint Mental Attitudes

#### Chapter 8. Organization-Oriented Systems

<table>
<thead>
<tr>
<th>Condition</th>
<th>Axiom</th>
</tr>
</thead>
<tbody>
<tr>
<td>( J_{O1} \subseteq { S(\alpha_1, \alpha_2) \cup J_{O3} } )</td>
<td>( S(\alpha_1, \alpha_2, \phi) \land J(\alpha_3, \phi) \supseteq J(\alpha_4, \phi) )</td>
</tr>
<tr>
<td><strong>Description:</strong> The situation in which a social mental attitude between two organizations and a joint mental attitude of one of the organizations lead to a joint mental attitude of one of the two organizations. An example of this situation is the authority of a commander to delegate goals. That is, when a commander has a joint goal then so will any organization that it commands: COMMAND((\omega, \omega', \phi) \land JGOAL(\omega, \phi) \supseteq JGOAL(\omega', \phi)).</td>
<td></td>
</tr>
<tr>
<td>( J_{O4} \cap { S(\alpha_1, \alpha_2) \cup J_{O3} } = \emptyset )</td>
<td>( S(\alpha_1, \alpha_2, \phi) \land J(\alpha_3, \phi) \supseteq \neg J(\alpha_4, \phi) )</td>
</tr>
<tr>
<td><strong>Description:</strong> The situation in which the existence of a social mental attitudes between two organizations and a joint mental attitude of one of the organizations towards a particular state formula implies that there are constraints on the joint mental attitude that one of these two organizations has towards the same state formula. An example of such a situation is the subordination of a commanded organization. That is, the commander has a joint goal towards a state formula but the commanded organization does not adopt a joint goal towards the same state formula: COMMAND((\omega, \omega', \phi) \land JGOAL(\omega, \phi) \supseteq \neg JGOAL(\omega', \phi)). Note that this does not imply that the commanded organization is obedient and actually adopts the joint goal of the commander.</td>
<td></td>
</tr>
<tr>
<td>( J_{O4} \cap { S(\alpha_1, \alpha_2) \cup J_{O3} } \neq \emptyset )</td>
<td>( S(\alpha_1, \alpha_2, \phi) \land J(\alpha_3, \phi) \supseteq \neg J(\alpha_4, \neg \phi) )</td>
</tr>
<tr>
<td><strong>Description:</strong> The situation in which the existence of a social mental attitudes between two organizations and a joint mental attitude of one of the organizations towards a particular state formula implies that there are constraints on the joint mental attitude that one of these two organizations has towards the negated state formula. An example of such a situation is the “loyalty” of a subordinate. That is, if a commanding organization has a joint goal towards a state formula then the commanded organization will not try to “sabotage” this goal, i.e., attempt to achieve the negation: COMMAND((\omega, \omega', \phi) \land JGOAL(\omega, \phi) \supseteq \neg JGOAL(\omega', \neg \phi)).</td>
<td></td>
</tr>
</tbody>
</table>

Table 8.3: Social and Joint Organization-Oriented Modal Systems (a)

cause itself to adopt a joint goal, joint intention, and mutual belief.

The semantic conditions that correspond to the above axioms are represented as multimodal containment conditions. That is the containment of one accessibility relation in another. More formally we have:

\[(CM{-}BC) \forall w \forall t \forall v \text{ if } (w, t, v) \in \bigcup_{CMI(\omega, \omega', w, t, v)} JG_{\omega'} \text{ then } (w, t, v) \in JG_{\omega};\]

\[(CT{-}BC) \forall w \forall t \forall v \text{ if } (w, t, v) \in \bigcup_{CTI(\omega, \omega', w, t, v)} JI_{\omega'} \text{ then } (w, t, v) \in JI_{\omega};\]

\[(CO{-}BC) \forall w \forall t \forall v \text{ if } (w, t, v) \in \bigcup_{COI(\omega, \omega', w, t, v)} MB_{\omega'} \text{ then } (w, t, v) \in MB_{\omega}.\]

We refer to an organization-oriented model in which the above semantic conditions hold as \(M^{ooos-b}\). Note that in this model the relationship that an organization has with other organizations (i.e., the social mental attitudes) impact on its internal mental state (i.e., the joint mental attitudes).

In the basic organization-oriented system there are no constraints on the relationship between the commanders and controllers of an organization. Intuitively this may lead to potential inconsistencies in the joint mental attitudes of an organization. That is, an organization is required to adopt a joint intention without having a corresponding joint goal. An example of such a situation in human organizations is when an adopted strategy determined by the CEO does not correspond to the adopted objectives determined by the Board of Directors. These inconsistencies arise only if we adopt a “strong realism” approach [88] to the rational behavior of an organization. That is, we require that joint
8.8. INTERACTION BETWEEN SOCIAL MENTAL ATTITUDES AND JOINT MENTAL ATTITUDES

Here we will prove the soundness and completeness of the three axioms for an organization-oriented model to take into account the new axioms introduced. Let us first define the corresponding tableau rules:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Axiom</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ J_{\alpha_5} \subseteq {S(\alpha_1, \alpha_2) \cup S(\alpha_3, \alpha_4) } ]</td>
<td>[ S(\alpha_1, \alpha_2, \phi) \land S(\alpha_3, \alpha_4, \phi) \supset J(\alpha_5, \phi) ]</td>
</tr>
</tbody>
</table>

**Description:** The situation in which two social mental attitudes imply the existence of a joint mental attitude. An example of such a situation is when social relationships with other organizations only exist as means for achieving a joint goal. For example if an organization has command and communication relationship with another organization towards a particular state formula then it has a joint goal to achieve this formula: \( \text{COMMAND}(\omega, \omega', \phi) \land \text{COMMS}(\omega, \omega, \phi) \supset J\text{GOAL}(\omega, \phi) \).

<table>
<thead>
<tr>
<th>Condition</th>
<th>Axiom</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ J_{\alpha_5} \cap {S(\alpha_1, \alpha_2) \cup S(\alpha_3, \alpha_4) } = \emptyset ]</td>
<td>[ S(\alpha_1, \alpha_2, \phi) \land S(\alpha_3, \alpha_4, \phi) \supset \neg J(\alpha_5, \phi) ]</td>
</tr>
</tbody>
</table>

**Description:** The situation in which two social mental attitudes imply the negation of a joint mental attitude. An example of such a situation is when social relationships with other organizations only exist as means for responding for a negated joint mental attitude, e.g., communication as a means for addressing lack of information as is the case in the staff information flow described above. That is, two organizations communicate with respect to a state formula only if one of the organizations does not already believe this state formula: \( \text{COMMS}(\omega, \omega, \phi) \land \text{COMMS}(\omega', \omega', \phi) \supset \neg \text{MBEL}(\omega, \phi) \).

<table>
<thead>
<tr>
<th>Condition</th>
<th>Axiom</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ J_{\alpha_5} \cap {S(\alpha_1, \alpha_2) \cup S(\alpha_3, \alpha_4) } \neq \emptyset ]</td>
<td>[ S(\alpha_1, \alpha_2, \phi) \land S(\alpha_3, \alpha_4, \phi) \supset \neg J(\alpha_5, \neg \phi) ]</td>
</tr>
</tbody>
</table>

**Description:** The situation in which two social mental attitudes towards a state formula imply the negation of a joint mental attitude towards the negated state formula. An example of such a situation is when social relationships with other organizations only exist as means for achieving a joint mental attitude the consistency between the social mental attitudes an the joint mental attitudes of one of the organizations, e.g., there are no “conflict of interest” of the commander and controller. That is, if an organization is both a commander and controller of another organization with respect to a particular state formula then it can not have a joint goal to the negation of that state formula: \( \text{COMMAND}(\omega, \omega', \phi) \land \text{CONTROL}(\omega, \omega', \phi) \supset \neg \text{JGOAL}(\omega, \neg \phi) \).

| Table 8.4: Social and Joint Organization-Oriented Modal Systems (b) |

Intentions are a sub-set of the joint goals. Alternatively we can adopt a “weak realism” approach [88], that is, joint intentions do not contradict goals. In this case there may not be any inconsistencies in the mental state of an organization.

8.8.2 Soundness and Completeness

Here we will prove the soundness and completeness of the three axioms (CM-BA), (CT-BA), and (CO-BA) with respect to the conditions (CM-BC), (CT-BC), and (CO-BC).

The \( \text{BGI}^{oos}_{\text{CTL}} \) axiom system and model described above are an extension to the organization-oriented model, \( \text{BGI}^{oos}_{\text{CTL}} \), described above. Here we will modify the proof of soundness and completeness for an organization-oriented model to take into account the new axioms introduced. Let us first define the corresponding tableau rules:

| \( \text{CM-BT} \) | If \( \text{JGOAL}(\omega, \phi) \in \Phi(w_t) \) and \( (w, t, v) \in \bigcup_{\text{CM}(\omega, \omega', w, t, v)} JG_{\omega'} \) then \( \forall \omega' : \text{COMMAND}(\omega, \omega', \phi) \in \Phi(w_t) \) we have \( \text{JGOAL}(\omega', \phi) \in \Phi(w_t) \) and \( \phi \in \Phi(v_t) \); |
| \( \text{CT-BT} \) | If \( \text{JINTEND}(\omega, \phi) \in \Phi(w_t) \) and \( (w, t, v) \in \bigcup_{\text{CT}(\omega, \omega', w, t, v)} JI_{\omega'} \) then \( \forall \omega' : \text{CONTROL}(\omega, \omega', \phi) \in \Phi(w_t) \) we have \( \text{JINTEND}(\omega', \phi) \in \Phi(w_t) \) and \( \phi \in \Phi(v_t) \); |
| \( \text{CO-BT} \) | If \( \text{MBEL}(\omega, \phi) \in \Phi(w_t) \) and \( (w, t, v) \in \bigcup_{\text{CO}(\omega, \omega', w, t, v)} MB_{\omega'} \) then \( \forall \omega' : \text{COMMS}(\omega, \omega', \phi) \in \Phi(w_t) \) we have \( \text{MBEL}(\omega', \phi) \in \Phi(w_t) \) and \( \phi \in \Phi(v_t) \); |
8.8. INTERACTION BETWEEN SOCIAL MENTAL ATTITUDES AND JOINT MENTAL ATTITUDES

CHAPTER 8. ORGANIZATION-ORIENTED SYSTEMS

<table>
<thead>
<tr>
<th>Condition</th>
<th>Axiom</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S(\alpha_1, \alpha_2) = {J_{\alpha_3} \cup J_{\alpha_4}} )</td>
<td>( J(\alpha_3, \phi) \land J(\alpha_4, \phi) \equiv S(\alpha_1, \alpha_2, \phi) )</td>
</tr>
<tr>
<td><strong>Description:</strong> The situation in which two joint mental attitudes of two organizations and a social mental attitude between those two organizations always exist together. An example of such a situation is the manifestation of the command relationship as a synchronization of the joint goals of the two organizations in the relationship. That is, when two organizations both have the same joint goal then there exist a command relationship between them and vice versa: ( \text{GOAL}(\omega, \phi) \land \text{GOAL}(\omega', \phi) \equiv \text{COMMAND}(\omega, \omega', \phi) ).</td>
<td></td>
</tr>
<tr>
<td>( S(\alpha_1, \alpha_2) \subseteq {J_{\alpha_3} \cup J_{\alpha_4}} )</td>
<td>( J(\alpha_3, \phi) \land J(\alpha_4, \phi) \supseteq S(\alpha_1, \alpha_2, \phi) )</td>
</tr>
<tr>
<td><strong>Description:</strong> The situation in which the social mental attitude emerges as a result of the synchronization of the joint mental attitudes of the two organizations. An example of such a situation is the synchronization of mutual beliefs through communications. That is, when two organizations both have mutual beliefs then this implies a communication relationship between them: ( \text{BEL}(\omega, \phi) \land \text{BEL}(\omega', \phi) \supseteq \text{COMMS}(\omega, \omega', \phi) ).</td>
<td></td>
</tr>
<tr>
<td>( S(\alpha_1, \alpha_2) \cap {J_{\alpha_3} \cup J_{\alpha_4}} \neq \emptyset )</td>
<td>( J(\alpha_3, \phi) \land J(\alpha_4, \phi) \supseteq \neg S(\alpha_1, \alpha_2, \neg \phi) )</td>
</tr>
<tr>
<td><strong>Description:</strong> The situation in which the synchronization of the joint mental attitudes of the two organizations towards a particular state formula implies a constraint on the social relationships towards the negated state formula. An example of such a situation is that there are no conflicts between the synchronization of joint intentions and the control relationship. That is, when two organizations both have joint intentions towards a particular state formula then this implies that there can not be a control relationship between them towards the negated state formula: ( \text{INTEND}(\omega, \phi) \land \text{INTEND}(\omega', \phi) \supseteq \neg \text{CONTROL}(\omega, \omega', \neg \phi) ).</td>
<td></td>
</tr>
</tbody>
</table>

Table 8.5: Social and Joint Organization-Oriented Modal Systems (c)

We can proceed by defining \( \text{BGI}^{\text{oos-b}}_{\text{CTL}} \)-tableau and pseudo-\( \text{BGI}^{\text{oos-b}}_{\text{CTL}} \)-tableau. We begin by defining a \( \text{BGI}^{\text{oos-b}}_{\text{CTL}} \)-tableau:

**Definition 27**

A \( \text{BGI}^{\text{oos-b}}_{\text{CTL}} \)-tableau is a \( \text{BGI}^{\text{os}}_{\text{CTL}} \)-tableau that also satisfies conditions (CM-BT), (CT-BT), and (CO-BT).

Similarly, one can define a pseudo-\( \text{BGI}^{\text{oos-b}}_{\text{CTL}} \)-tableau. We refer to the class of structures \( \mathcal{M}^{\text{oos-b}} \) that satisfies the multi-modal containment conditions (MB-BC), (JG-BC), and (JI-BC) as \( \mathcal{M}^{\text{oos-b}} \). Let us now prove the small model theorem for \( \text{BGI}^{\text{oos-b}}_{\text{CTL}} \).

**Theorem 13** Let \( \phi_0 \) be a \( \text{BGI}_{\text{CTL}} \) formula of length \( n \), then the following equivalences hold:

1. \( \phi_0 \) is \( \text{BGI}^{\text{oos-b}}_{\text{CTL}} \)-satisfiable.;
2. \( \phi_0 \) has a model \( \mathcal{M}^{\text{oos-b}} \) with a finite branching in each world bounded by \( \mathcal{O}(n) \);
3. \( \phi_0 \) has a finite pseudo-\( \text{BGI}^{\text{oos-b}}_{\text{CTL}} \)-tableau of size \( \leq \exp(n) \);
4. \( \phi_0 \) has a finite model \( \mathcal{M}^{\text{oos-b}} \) of size \( \leq \exp(n) \).

**Proof:**

For the CTL component the proof is identical to the proof of Theorem 11. For the BGI component we prove the equivalence between the semantic constraints (CM-BC),

© Copyright Gil Tidhar, 1999 159
(CT-BC), and (CO-BC), and the tableau rules (CM-BT), (CT-BT), and (CO-BT), respectively. We will proceed to show that (CO-BC) is equivalent to (CO-BT). The proof for the other equivalences is similar.

1. For one direction, given (CO-BC), that is \( \forall \omega \forall t \forall v \) if \( (w, t, v) \in \bigcup_{\omega \in \mathcal{CO}(\omega, \omega', w, t, v)} MB_{\omega'} \) then \( (w, t, v) \in MB_{\omega} \), we want to prove that (CO-BT) holds, that is if \( \mathrm{MBEL}(\omega, \phi) \in \Phi(w_t) \) and \( (w, t, v) \in \bigcup_{\omega \in \mathcal{CO}(\omega, \omega', w, t, v)} MB_{\omega'} \) then \( \forall \omega': \mathrm{COMMS}(\omega, \omega', \phi) \in \Phi(w_t) \) we have \( \mathrm{MBEL}(\omega, \phi) \in \Phi(w_t) \).

We first show that \( \phi \in \Phi(v_t) \). Given that \( (w, t, v) \in \bigcup_{\omega \in \mathcal{CO}(\omega, \omega', w, t, v)} MB_{\omega'} \) we get from the assumption that \( (w, t, v) \in MB_{\omega} \). It follows that given that \( \mathrm{MBEL}(\omega, \phi) \in \Phi(w_t) \) we get from the BGI^oo-\text{consistency} rules that \( \phi \in \Phi(v_t) \).

We now need to show that it also holds that \( \forall \omega': \mathrm{COMMS}(\omega, \omega', \phi) \in \Phi(w_t) \) we have \( \mathrm{MBEL}(\omega', \phi) \in \Phi(v_t) \). Without loss of generality we will consider a particular organization \( \omega_k \) such that \( \mathrm{COMMS}(\omega, \omega_k, \phi) \in \Phi(w_t) \). We will assume that \( \mathrm{MBEL}(\omega_k, \phi) \notin \Phi(v_t) \) and show a contradiction.

Based on the assumption and the BGI^oo-\text{consistency} rules we get that there exists \( v' \) such that \( (w, t, v') \in MB_{\omega_k} \) and \( \phi \notin \Phi(v_t) \).

Given that \( \mathrm{COMMS}(\omega, \omega_k, \phi) \in \Phi(w_t) \) we get from the definition of the \( \mathrm{COMMS} \) operator that \( \mathcal{CO}(\omega, \omega_k, w, t, v') \). Given that \( (w, t, v') \in MB_{\omega_k} \) we get from the assumed rule that \( (w, t, v') \in MB_{\omega} \). From the first part of this proof we get that it holds that \( \phi \in \Phi(v_t) \) which contradicts our findings that \( \phi \notin \Phi(v_t) \).

2. For the reverse direction, given (CO-BT), that is if \( \mathrm{MBEL}(\omega, \phi) \in \Phi(w_t) \) and \( (w, t, v) \in \bigcup_{\omega \in \mathcal{CO}(\omega, \omega', w, t, v)} MB_{\omega'} \) then \( \forall \omega': \mathrm{COMMS}(\omega, \omega', \phi) \in \Phi(w_t) \) we have \( \mathrm{MBEL}(\omega', \phi) \in \Phi(v_t) \), we want to prove that (CO-BC) holds, that is \( \forall \omega \forall t \forall v \) if \( (w, t, v) \in \bigcup_{\omega \in \mathcal{CO}(\omega, \omega', w, t, v)} MB_{\omega'} \) then \( (w, t, v) \in MB_{\omega} \).

Without loss of generality we will consider a particular organization \( \omega_j \), worlds \( w' \) and \( v' \), time point \( t' \), and proposition \( \phi' \). We assume that \( \mathrm{MBEL}(\omega_j, \phi') \in \Phi(v_t) \) and \( (w', t', v') \in \bigcup_{\omega \in \mathcal{CO}(\omega_j, \omega', w', t', v')} MB_{\omega'} \). It follows that \( \forall \omega': \mathrm{COMMS}(\omega_j, \omega', \phi') \in \Phi(w_t) \) and also that \( \phi' \in \Phi(v_t) \). We need to show that \( (w', t', v') \in MB_{\omega_j} \). We will assume that \( (w', t', v') \notin MB_{\omega_j} \) and show a contradiction.

If \( (w', t', v') \notin MB_{\omega_j} \) then \( \mathrm{MBEL}_{\omega_j}([w'_t]) \not\subseteq [v'_t] \). In other words, there exists \( \psi \) such that \( \psi \in \mathrm{MBEL}_{\omega_j}([w'_t]) \) and \( \psi \notin [v'_t] \).

As \( (w', t', v') \in \bigcup_{\omega \in \mathcal{CO}(\omega_j, \omega', w', t', v')} MB_{\omega'} \) there exists an organization \( \omega_k \) such that \( \mathcal{CO}(\omega_j, \omega_k, w', t', v') \) holds, and for \( \omega_k \) we have \( (w', t', v') \in MB_{\omega_k} \) and \( \mathrm{MBEL}_{\omega_k}([w'_t]) \subseteq [v'_t] \). Given that \( \psi \in \mathrm{MBEL}_{\omega_j}([w'_t]) \) then from the definition we get that \( \mathrm{MBEL}(\omega_j, \psi) \in \Phi(w_t) \) and from our assumption we get that for \( \omega_k \) we have \( \mathrm{MBEL}(\omega_k, \psi) \in \Phi(w_t) \). It follows that \( \psi \in \mathrm{MBEL}_{\omega_k}([w'_t]) \). Given that \( \mathrm{MBEL}_{\omega_k}([w'_t]) \subseteq [v'_t] \) we get that \( \psi \in [v'_t] \), which contradicts our findings that \( \psi \notin [v'_t] \).
Let us now describe the algorithm for constructing a pseudo-BGI$_{\text{CTL}}$-tableau. Again the algorithm is essentially the same algorithm described above with additional steps added after Step 2(d) to take into account the additional constraints. These steps ensure that when a node includes a joint mental attitude and social mental attitude then the relevant nodes are added to ensure that the respective joint mental attitude of the organization in the respective social relationship are considered. The additional step, referred to as 2(d') is as follows:

2(d')

1. If $\Phi(w_t)$ contains $\text{JGOAL}(\omega, \phi)$ then $\forall \omega'$ such that $\Phi(w_t)$ contains $\text{COMMAND}(\omega, \omega', \phi)$, then let $\Phi(w_t) = \Phi(w_t) \cup \{ \text{JGOAL}(\omega', \phi) \}$ and $\forall v$ such that $(w, t, v) \in \mathcal{J}_\omega$, let $\Phi(v_t) = \Phi(v_t) \cup \{ \phi \}$. If there is no such $v$ then create one and initialize as above.

2. If $\Phi(w_t)$ contains $\text{JINTEND}(\omega, \phi)$ then $\forall \omega'$ such that $\Phi(w_t)$ contains $\text{CONTROL}(\omega, \omega', \phi)$, then let $\Phi(w_t) = \Phi(w_t) \cup \{ \text{JINTEND}(\omega', \phi) \}$ and $\forall v$ such that $(w, t, v) \in \mathcal{J}_\omega$, let $\Phi(v_t) = \Phi(v_t) \cup \{ \phi \}$. If there is no such $v$ then create one and initialize as above.

3. If $\Phi(w_t)$ contains $\text{MBEL}(\omega, \phi)$ then $\forall \omega'$ such that $\Phi(w_t)$ contains $\text{COMMS}(\omega, \omega', \phi)$, then let $\Phi(w_t) = \Phi(w_t) \cup \{ \text{MBEL}(\omega', \phi) \}$ and $\forall v$ such that $(w, t, v) \in \mathcal{M}_\omega$, let $\Phi(v_t) = \Phi(v_t) \cup \{ \phi \}$. If there is no such $v$ then create one and initialize as above.

4. If the label of a new leaf node $w_t$ is identical to the label of an ancestral node, then erase node $w_t$.  

$c^\odot$ Copyright Gil Tidhar, 1999 161
Consider the tableau illustrated in Figure 8.7. We start with the same formulas as in Figure 8.6 (Section 8.8.2). Note the effect of adding Step 2(d') - node M1 is blatantly inconsistent because it now includes both the propositions $p$ and $\neg p$. Similarly for node M0 - it now includes the formulas $\text{JGOAL}(b, p)$ and $\neg \text{JGOAL}(b, p)$. We are now in a position to prove that soundness and completeness of the $\text{BGIO}^{\text{tost}}$-system with respect to $\mathcal{M}^{\text{tost}}$. We are now in a position to prove that soundness and completeness of the $\text{BGIO}^{\text{tos}}$-system with respect to $\mathcal{M}^{\text{tos}}$.

**Theorem 14** The $\text{BGIO}^{\text{tos}}$-system is a sound and complete axiomatization with respect to $\mathcal{M}^{\text{tos}}$.

**Proof:**

To show that $\text{BGIO}^{\text{tos}}$-system is sound and complete we need to show that a formula $\phi_0$ is $\text{BGIO}^{\text{tos}}$-provable if and only if $\phi_0$ is marked ‘satisfiable’ in a pseudo-$\text{BGIO}^{\text{tos}}$-tableau. The proof is essentially identical to the proof provided above for Theorem 12.

We need to show that the axioms (CM-BA), (CT-BA), and (CO-BA) are equivalent to the tableau rules (CM-BT), (CT-BT), and (CO-BT) respectively. We will show that (CO-BA) is equivalent to (CO-BT). The other equivalences can be shown in a similar way.

1. For one direction, given (CO-BA), that is $\text{MBEL}(\omega, \phi) \wedge \text{COMMS}(\omega, \omega', \phi) \supset \text{MBEL}(\omega', \phi)$, we want to prove that (CO-BT) holds, that is if $\text{MBEL}(\omega, \phi) \in \Phi(w_t)$ and $(w, t, v) \in \bigcup_{\omega} \text{MB}_{\omega}$ then $\forall \omega': \text{COMMS}(\omega, \omega', \phi) \in \Phi(w_t)$ we have $\text{MBEL}(\omega', \phi) \in \Phi(w_t)$ and $\phi \in \Phi(v_t)$.

Without loss of generality we will consider particular organizations $\omega_j$ and $\omega_k$, world $w'$ and $v'$, and time point $t'$, it holds that $\text{MBEL}(\omega_j, \phi) \in \Phi(w_j')$, $\text{CO}(\omega_j, \omega_k, w', t', v')$ holds, $(w', t', v') \in \text{MB}_{\omega_k}$, and show that $\text{MBEL}(\omega_k, \phi) \in \Phi(w_k')$ and $\phi \in \Phi(v_k')$.

Let us first show that $\text{MBEL}(\omega_k, \phi) \in \Phi(w_k')$. Given that $\text{MBEL}(\omega_j, \phi) \in \Phi(w_j')$ then from the quotient construction we get that $M^q, [w_j'] \models \text{MBEL}(\omega_j, \phi)$. Given that we assumed that $\text{CO}(\omega_j, \omega_k, w', t', v')$ holds, and hence $\text{CO}^q(\omega_j, \omega_k, [w_j'])$ holds, we get that $\text{COMMS}(\omega_j, \omega_k, \phi) \in \Phi(w_j')$. From the quotient construction we get that $M^q, [w_j'] \models \text{COMMS}(\omega_j, \omega_k, \phi)$.

From (MB-EA) we now get that $M^q, [w_j'] \models \text{MBEL}(\omega_k, \phi)$. It follows that $\text{MBEL}(\omega_k, \phi) \in \Phi(w_j')$ as required.

We will now show that $\phi \in \Phi(v_j')$. Note that we have assumed that $(w', t', v') \in \text{MB}_{\omega_k}$ and that we have shown above that $\text{MBEL}(\omega_k, \phi) \in \Phi(w_j')$. It follows from the $\text{BGIO}^{\text{tos}}$-consistency rules that $\phi \in \Phi(v_j')$ as required.

2. For the reverse direction, given (CO-BT), that is if $\text{MBEL}(\omega, \phi) \in \Phi(w_t)$ and $(w, t, v) \in \bigcup_{\omega} \text{MB}_{\omega}$ then $\forall \omega': \text{COMMS}(\omega, \omega', \phi) \in \Phi(w_t)$ we have $\text{MBEL}(\omega', \phi) \in \Phi(w_t)$ and $\phi \in \Phi(v_t)$ as required.
8.8. INTERACTION BETWEEN SOCIAL MENTAL ATTITUDES AND JOINT MENTAL ATTITUDES

\[ \Phi(v_t) \text{ and } \phi \in \Phi(v_t), \text{ we want to prove that (MB-EA) holds, that is } \text{MBEL}(\omega, \phi) \land \text{COMMS}(\omega, \omega', \phi) \supset \text{MBEL}(\omega', \phi). \]

Without loss of generality we will consider particular organizations \( \omega_j \) and \( \omega_k \), world \( \omega' \), time point \( t' \), such that \( \text{MBEL}(\omega_j, \phi) \) and \( \text{COMMS}(\omega_j, \omega_k, \phi) \) holds. We will show that \( \text{MBEL}(\omega_k, \phi) \) holds.

Given that \( \text{MBEL}(\omega_j, \phi) \) holds then we have that \( M^q, [w_t'] \models \text{MBEL}(\omega_j, \phi) \). It follows that \( \text{MBEL}(\omega_j, \phi) \in \Phi(w_t') \). Similarly we can show that \( \text{COMMS}(\omega_j, \omega_k, \phi) \in \Phi(w_t') \).

It follows that \( \phi \in \text{COMMS}^- (\omega_j, \omega_k, [w_t']) \) and hence \( \text{CO}(\omega_j, \omega_k, [w_t']) \) holds. It follows that \( CO(\omega_j, \omega_k, w', t', t'') \) holds.

From the tableau rule we get that \( \text{MBEL}(\omega_k, \phi) \in \Phi(w_t') \). It follows that \( M^q, [w_t'] \models \text{MBEL}(\omega_k, \phi) \). We thus get that \( \text{MBEL}(\omega_k, \phi) \) holds.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Axiom</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S(\omega, \omega') \subseteq ST(\omega, \omega') )</td>
<td>( \text{SUBTEAM}(\omega, \omega') \supset S(\omega, \omega'), \phi) )</td>
</tr>
<tr>
<td>Description: The situation in which a subteam relation implies the existence of a social relation between the two organizations. An example of such a situation is that an organization can command its sub-tasks as is sometimes implied when describing an organizational chart: ( \text{SUBTEAM}(\omega, \omega') \supset \text{COMMAND}(\omega, \omega', \phi) ).</td>
<td></td>
</tr>
<tr>
<td>( S(\omega, \omega') \cap ST(\omega, \omega') = \emptyset )</td>
<td>( S(\omega, \omega'), \phi) \supset \neg \text{SUBTEAM}(\omega, \omega') )</td>
</tr>
<tr>
<td>Description: The situation in which there is a separation between the social mental attitudes and the team structure. An example of such a situation is communication that only flows outside the organization. That is, an organization only has communication relationships with organizations that are not its sub-tasks: ( \text{COMMS}(\omega, \omega'), \phi) \supset \neg \text{SUBTEAM}(\omega, \omega') ).</td>
<td></td>
</tr>
<tr>
<td>( S_2(\alpha_5, \alpha_6) \subseteq {S_1(\alpha_1, \alpha_2) \cup ST(\alpha_3, \alpha_4)} )</td>
<td>( S_1(\alpha_1, \alpha_2, \phi) \land \text{SUBTEAM}(\alpha_3, \alpha_4) \supset S_2(\alpha_5, \alpha_6, \phi) )</td>
</tr>
<tr>
<td>Description: The situation in which a social relationship and subteam relationship implies another social relationship. An example of such a situation is when a commander does not engage in “micro-management”. That is, a commander of an organization can command any of the sub-tasks of the commanded organization: ( \text{COMMAND}(\omega, \omega', \phi) \land \text{SUBTEAM}(\omega', \omega'') \supset \text{COMMAND}(\omega, \omega'', \phi) ). Another example is the requirement of all sub-tasks to communicate with the commander of the whole organization: ( \text{COMMAND}(\omega, \omega', \phi) \land \text{SUBTEAM}(\omega', \omega'') \supset \text{COMMAND}(\omega, \omega'', \phi) ).</td>
<td></td>
</tr>
<tr>
<td>( S_2(\alpha_5, \alpha_6) \cap {S_1(\alpha_1, \alpha_2) \cup ST(\alpha_3, \alpha_4)} = \emptyset )</td>
<td>( S_1(\alpha_1, \alpha_2, \phi) \land \text{SUBTEAM}(\alpha_3, \alpha_4) \supset \neg S_2(\alpha_5, \alpha_6, \phi) )</td>
</tr>
<tr>
<td>Description: The situation in which the existence of a social mental attitude between two organizations implies a restriction on the social mental attitudes that exist between the first organization and the sub-team. An example of such a situation is when a commander does not engage in “micro-management”. That is, a commander does not control the sub-tasks of the commanded organization: ( \text{COMMAND}(\omega, \omega', \phi) \land \text{SUBTEAM}(\omega', \omega'') \supset \neg \text{CONTROL}(\omega, \omega'', \phi) ). Another example is the independence of the commander from influences by any of the sub-tasks it commands (i.e., a strict command hierarchy): ( \text{COMMAND}(\omega, \omega', \phi) \land \text{SUBTEAM}(\omega', \omega'') \supset \neg \text{COMMAND}(\omega, \omega'', \phi) ).</td>
<td></td>
</tr>
<tr>
<td>( S_2(\alpha_5, \alpha_6) \cap {S_1(\alpha_1, \alpha_2) \cup ST(\alpha_3, \alpha_4)} \neq \emptyset )</td>
<td>( S_1(\alpha_1, \alpha_2, \phi) \land \text{SUBTEAM}(\alpha_3, \alpha_4) \supset \neg S_2(\alpha_5, \alpha_6, \neg \phi) )</td>
</tr>
<tr>
<td>Description: The situation in which a social mental attitude between two organizations with respect to a particular state formula implies consistency with the social mental attitudes between the first organization and the sub-tasks of the second organization. An example of such a situation is that the authority to command an organization as to ( \phi ) and the ability to command a sub-team of the organization as to ( \neg \phi ): ( \text{COMMAND}(\omega, \omega', \phi) \land \text{SUBTEAM}(\omega', \omega'') \supset \neg \text{COMMAND}(\omega, \omega'', \phi) ). Another example is the corresponding situation: ( \text{COMMAND}(\omega, \omega', \phi) \land \text{SUBTEAM}(\omega', \omega'') \supset \neg \text{COMMAND}(\omega, \omega', \neg \phi) ).</td>
<td></td>
</tr>
</tbody>
</table>

Table 8.6: Social and Structure Organization-Oriented Modal Systems
8.9 Interaction Between Social Mental Attitudes and Team Structure

In this section we consider the possible interactions between the social mental attitudes and the structure of the organizations (i.e., team structure) in the social relationship. The team-structure of an organization is represented through the sub-team relation (i.e., $ST$) and the corresponding operator (i.e., $\text{SUBTEAM}$). Recall that the $ST(\omega, \omega')$ relation is defined as $\mathcal{MB}_\omega \cup \mathcal{MB}_{\omega'}$.

We consider two types of interactions between the social mental attitudes and the team-structure of an organization (Table 8.6). The first describes the possible relations between a single social mental attitude and the team structure. The second involves two social mental attitudes and the team structure. As is the case with the interactions between different social mental attitudes these three relations may involve either two or three organizations. In Table 8.6 we use $\alpha_i$, where $1 \leq i \leq 6$, as place holders in which any of $\omega$ or $\omega'$ could be placed to form different types of conditions and axioms.

8.10 A Model of Organizational Structures

In his work on formal organizations Kuhn [70, pages 227–228] defines a structure for an organization to consist of its division into sub-organizations and the functions performed by and relationships between these sub-organizations. In Section 8.4 we defined the team structure through the specification of the sub-team relationship. In the previous section we provided a formal definition of the relationships that exist between organizations through the specification of social mental attitudes.

We now define an organizational structure for an organization to include a set of sub-team relations for this organization and a set of social mental attitudes between the organization and the sub-teams or between two sub-teams. More formally:

**Definition 28**

Given an organization $\omega$, an *organizational structure* for $\omega$ is a pair $\sigma = \langle S_t, S_s \rangle$. Where $S_t$ is a set of sub-team relations that $\omega$ has with other organizations (i.e., relations of the form $\text{SUBTEAM}(\omega, \omega_i)$ where $\omega_i$ is a sub-team of $\omega$). $S_s$ is a set of social mental attitudes between $\omega$ and any of the sub-teams $\omega_i$ or between two sub-teams. This social mental attitude is with respect to any state formula. We refer to $S_s$ as the *social structure* for $\omega$.

Given a state formula $\phi$ we can consider the social mental attitudes in a social structure $S_s$ with respect to $\phi$. We can view the social structure as a combination of sets of social mental attitudes where each set is associated with a different state formula. According to our definition each of these sets is also a social structure for the organization. This approach will be discussed when describing the operational semantics in Chapter 10.
system and the relevant axioms adopted. A discussion on the operational semantics of organizational structures for the basic organization-oriented system is provided in Chapter 10.
Chapter 9

Axiom Systems for Organization-Oriented Systems

Let us now consider a variety of axiom systems that determine the interaction between the different social mental attitudes that exist between two organizations and the effects on the social mental attitudes, joint mental attitudes, and structure of the two organizations.

The organization-oriented systems described below are representative of human organizations and social phenomena that can be expressed using our framework. They also represent different types of interactions between the social mental attitudes, joint mental attitudes, and team structure. We present the following organization-oriented systems:

**Dominant Coalition:** A dominant coalition is identified by the existence of a subset of the sub-teams of an organization that are dominant in determining the joint goals, joint intentions, and mutual beliefs of the whole organization.

**Military Unit:** A military unit is identified by a very small dominant coalition, i.e., the commander of the unit, and a clear chain of command.

**Simple Organization:** A simple organization is identified by the centralization of decisions and direct communications between the single decision maker and every member of the organization.

**Divisionalized Organization:** A divisionalized organization is identified by a set of quasi-autonomous organizations (i.e., the divisions) and a central strategic apex (i.e., the headquarters) [79]. The headquarters is responsible for setting high-level objectives for the organization, establishing high level plans for the organization, setting objectives for the divisions, monitoring the performance of the divisions, and adding or removing divisions.
9.1 The Dominant Coalition

In the basic organization oriented system described in Chapter 8 we have shown how social mental attitudes lead to joint mental attitude. Recall that in the BGI_{\text{os-e}}^{\text{CTL}} system we required that if an organization has a joint mental attitude then so will all its sub-teams. This approach did not distinguish between the various sub-teams of an organization. In the organization-oriented approach we can also constrain the joint mental attitudes by the existing social mental attitudes.

The behavior of an organization is determined by its joint intentions. These in turn are determined by its mutual beliefs and joint goals. When considering social mental attitudes one has to decide on the way they affect the joint mental attitudes. One approach is to formalize the notion of “dominant coalition” in Organization Theory as defined by Cyert and March [23]. A dominant coalition is defined to be a subset of the members of the organization that are dominant in determining the objectives, plans, and decisions of the whole organization.

Here we consider an axiom system that has the organizations that are in a command relationship being considered (i.e., the “commanders”) as dominant in the adoption of the joint mental attitudes of the whole organization. In such a system only the commanders of the organization are considered for the purpose of defining the joint mental attitudes adopted by the organization. That is, we say that if an organization has a joint goal, joint intention, or mutual belief then so will all the organizations that command it.

The axioms that reflect the dominant coalition approach are similar to the every sub-team axioms described in Section 7.1.1. More formally we have the axiom schemas:

\[(\text{MB-CMA}) \quad \text{MBEL}(\omega', \phi) \land \text{COMMAND}(\omega, \omega', \phi) \supset \text{MBEL}(\omega, \phi)\]

\[(\text{JG-CMA}) \quad \text{JGOAL}(\omega', \phi) \land \text{COMMAND}(\omega, \omega', \phi) \supset \text{JGOAL}(\omega, \phi)\]

\[(\text{JI-CMA}) \quad \text{JINTEND}(\omega', \phi) \land \text{COMMAND}(\omega, \omega', \phi) \supset \text{JINTEND}(\omega, \phi)\]

In the above approach we require that only the mutual beliefs, joint goals, and joint intentions of the commanding organizations be consistent with the attitudes of the whole organization. We denote an organization-oriented system in which the above axioms hold as the BGI_{\text{os-dc}}^{\text{CTL}} system.

Note that from a practical point of view one can consider this as a form of rational model of organizations [1]. That is, limiting the number of organizations that determine the joint mental attitudes of an organization reduces the amount of synchronization of mental states that is required. This approach is brought to an extreme in a military unit where each unit would have only one or two commanding officers.

The semantic condition that enforces the above axiom is similar to the (MB-EC), (JG-EC), and (JI-EC) conditions described in Section 7.1.2. That is, the mutual-belief, joint-goal, and joint-intention-accessible worlds of an organization contain the union of the respective mutual-belief, joint-goal, and joint-intention-accessible worlds of the command-
9.2. MILITARY AXIOM SYSTEMS FOR ORGANIZATION-ORIENTED SYSTEMS

ing organizations. We thus require that for every organization $\omega'$ the following rules hold:

\[ (\text{MB-CMC}) \forall w \forall t \forall v \text{ if } (w, t, v) \in MB_\omega \text{ then } (\omega, \omega', w, t, v) \in CM \text{ or } (w, t, v) \in MB_{\omega'}; \]

\[ (\text{JG-CMC}) \forall w \forall t \forall v \text{ if } (w, t, v) \in JG_\omega \text{ then } (\omega, \omega', w, t, v) \in CM \text{ or } (w, t, v) \in JG_{\omega'}; \]

\[ (\text{JI-CMC}) \forall w \forall t \forall v \text{ if } (w, t, v) \in JI_\omega \text{ then } (\omega, \omega', w, t, v) \in CM \text{ or } (w, t, v) \in JI_{\omega'} . \]

We refer to an organization-oriented model in which the above semantic conditions hold as $M_{\text{mos–dc}}$.

9.2 A Military Unit

One of the most formal organizations is the military unit. We will now provide an organization-oriented system that follows the model of a military unit. The principles that lead to a military unit are the “dominant coalition”, “chain of command”, and some constraints on the control and communication relationships.

The chain of command is represented through the transitivity of the social mental attitudes relationship with respect to different organizations and with respect to the sub-team relation. The constraints on the control and communication relations require that the commander is also the controller and that the commander communicates with the subordinates and vice versa. There are no requirements for communication between the subordinates.

9.2.1 Transitivity of the Social Mental Attitudes

The chain of command is represented through the transitivity of the command, control, and communication relationships. That is, if (1) an organization $\omega$ has a command relationship with another organization $\omega'$; and (2) $\omega'$ has a command relationship with a third organization $\omega''$; then $\omega$ has a command relationship with $\omega''$. Similarly for control and communication relationships. More formally:

\[ (\text{CM-TA}) \text{ COMMAND}(\omega, \omega', \phi) \land \text{ COMMAND}(\omega', \omega'', \phi) \supset \text{ COMMAND}(\omega, \omega'', \phi) \]

\[ (\text{CM-TA}) \text{ CONTROL}(\omega, \omega', \phi) \land \text{ CONTROL}(\omega', \omega'', \phi) \supset \text{ CONTROL}(\omega, \omega'', \phi) \]

\[ (\text{CO-TA}) \text{ COMMS}(\omega, \omega', \phi) \land \text{ COMMS}(\omega', \omega'', \phi) \supset \text{ COMMS}(\omega, \omega'', \phi) \]

We refer to an organization-oriented system with the above axioms as the BGI$_{\text{mos–ctl}}$ system. The semantic conditions that enforce the above model are based on a relationship between the accessibility relations of the organizations and the accessibility relations of the
sub-teams. We require that for all organizations $\omega$, $\omega'$, and $\omega''$ the following condition hold:

\[(CM-TC) \forall w \forall t \forall v \text{ if } (\omega, \omega'', w, t, v) \in CM \text{ then } (\omega, \omega', w, t, v) \in CM \text{ or } (\omega', \omega'', w, t, v) \in CM;\]

\[(CT-TC) \forall w \forall t \forall v \text{ if } (\omega, \omega'', w, t, v) \in CT \text{ then } (\omega, \omega', w, t, v) \in CT \text{ or } (\omega', \omega'', w, t, v) \in CT;\]

\[(CO-TC) \forall w \forall t \forall v \text{ if } (\omega, \omega'', w, t, v) \in CO \text{ then } (\omega, \omega', w, t, v) \in CO \text{ or } (\omega', \omega'', w, t, v) \in CO.\]

We refer to an organization-oriented model in which the above semantic conditions hold as $M^{oo-o}$.

### 9.2.2 Sub-team Transitivity of the Social Mental Attitudes

In the $BGI^{oo-o}_{CTL}$ system an organization that is in a command relationship with another organization does not necessarily have a relationship with any of the sub-teams of that organization. That is, it may not be able to cause a sub-team of an organization it commands to adopt a goal. To allow the commanding organization to command a sub-team we add an axiom in which if an organization has a command relationship with an organization then it also has a command relationship with all its sub-teams. Similar axioms are provided for control and communication. More formally:

\[(CM-STA) \text{ COMMAND}(\omega, \omega', \phi) \land \text{ SUBTEAM}(\omega', \omega'') \supset \text{ COMMAND}(\omega, \omega'', \phi);\]

\[(CM-STA) \text{ CONTROL}(\omega, \omega', \phi) \land \text{ SUBTEAM}(\omega', \omega'') \supset \text{ CONTROL}(\omega, \omega'', \phi);\]

\[(CO-STA) \text{ COMMS}(\omega, \omega', \phi) \land \text{ SUBTEAM}(\omega', \omega'') \supset \text{ COMMS}(\omega, \omega'', \phi).\]

We refer to an organization-oriented system with the above axioms as the $BGI^{oo-o}_{CTL}$ system. The semantic conditions that enforce the above model are based on a relationship between the accessibility relations of the organizations and the accessibility relations of the sub-teams. We require that for all organizations $\omega$, $\omega'$, and $\omega''$ the following condition hold:

\[(CM-STC) \forall w \forall t \forall v \text{ if } (\omega, \omega'', w, t, v) \in CM \text{ then } (\omega', \omega'', w, t, v) \in ST \text{ or } (\omega, \omega', w, t, v) \in CM;\]

\[(CT-STC) \forall w \forall t \forall v \text{ if } (\omega, \omega'', w, t, v) \in CT \text{ then } (\omega', \omega'', w, t, v) \in ST \text{ or } (\omega, \omega', w, t, v) \in CT;\]

\[(CO-STC) \forall w \forall t \forall v \text{ if } (\omega, \omega'', w, t, v) \in CO \text{ then } (\omega', \omega'', w, t, v) \in ST \text{ or } (\omega, \omega', w, t, v) \in CO.\]
We refer to an organization oriented model in which the above semantic conditions hold as $M^{oos-st}$.

### 9.2.3 Constraints on Control and Communication

The constraint on the control relationships is that the commander of an organization is also the controller of that organization. That is, a command relationship implies a control relationship. More formally:

\[ (CT-CTA) \, \text{COMMAND}(\omega, \omega', \phi) \supset \text{CONTROL}(\omega, \omega', \phi) \]

The semantic conditions that enforce the above model are based on a relationship between the command accessibility relation and the control accessibility relation. We require that for all organizations $\omega$ and $\omega'$ the following condition hold:

\[ (CT-CTC) \, \forall w \forall t \forall v \text{ if } (\omega, \omega', w, t, v) \in CT \text{ then } (\omega, \omega', w, t) \in CM \]

The constraint on the communication relationships is that the commander of an organization communicates with the organizations it commands and that these organizations communicate with it. That is, a command relationship implies the respective communication relationships. More formally:

\[ (CO-CCA) \, \text{COMMAND}(\omega, \omega', \phi) \supset \text{COMMS}(\omega, \omega', \phi) \land \text{COMMS}(\omega', \omega, \phi) \]

The semantic conditions that enforce the above model are based on a relationship between the command accessibility relation and the control accessibility relation. We require that for all organizations $\omega$ and $\omega'$ the following condition hold:

\[ (CO-CCC) \, \forall w \forall t \forall v \text{ if } (\omega, \omega', w, t, v) \in CO \text{ or } (\omega', \omega, w, t, v) \in CO \text{ then } (\omega, \omega', w, t) \in CM \]

We refer to an organization-oriented system with the above axioms as the BGI$^{oos-ct}$ system. We refer to an organization oriented model in which the above semantic conditions hold as $M^{oos-ct}$.

### 9.3 The Simple Organization

Recall the simple organization described in Chapter 8. In axiomatization of this organization we identified axioms that describe the relationship between different social mental attitudes and between these attitudes and the team structure. These axioms were:
9.4 The Divisionalized Organization

The divisionalized organization is composed of a set of quasi-autonomous organizations (i.e., the divisions) and a central strategic apex (i.e., the headquarters) [79]. The headquarters is primarily responsible for establishing strategic plans for the organization, setting high-level objectives for the divisions, monitoring the performance of the divisions, and adding...
9.4. THE DIVISIONALIZED ORGANIZATION

Each division operates autonomously to achieve the objectives set by the headquarters. It is responsible for developing and executing plans that are designed to achieve the objectives. It is also responsible for reporting to the headquarters on its performance. The form of reporting is standardized across the divisions.

The structure of the headquarters would typically include a president and department that deal with strategic planning, e.g., business planning, legal issues, finance, etc.

The structure of each division can be anything from a simple structure, as is the case in a chain of newspaper agencies, to a military unit as is the case in the Roman Empire. Mintzberg claims that the divisionalized organization works best when the divisions adopt a structure with close control throughout the division (referred to as the machine bureaucracy) [79, pp. 385]. It seems then that the divisionalized organization is composed of three types of organizational structures:

The **Headquarters**: has a relatively flat structure with command, control, and communication relationships shared amongst the members;

The **Divisions**: have their own internal structure but typically there is centralized command and control in the hands of the manager of the division; and

The **Divisionalized Organization**: is commanded and controlled by the headquarters but the relationship between the divisionalized organization and the divisions only includes a command and communication relationships. The divisions are independent to determine the means of achieving the goals delegated to them and as such they are not controlled by the divisionalized organization.

Let us now describe the set of axioms that will reflect the relationship between the headquarters and the divisionalized organization and between the divisionalized organization and the various divisions. The model of this organization starts with the basic organization-oriented model. That is axioms (CM-BA), (CT-BA), and (CO-BA).

Let $\omega_{do}$ be the divisionalized organization, $\omega_{hq}$ the headquarters, and $\omega_1, \ldots, \omega_n$ the divisions. We thus get that $\omega_{hq}$ has a command, control, and communication relationships with $\omega_d$. The divisionalized organization $\omega_{do}$ only has a command relationship with the divisions. The divisions on the other hand only have a communication relationship with $\omega_{do}$. These relationships carry across the structure to the headquarters and can be modelled as the following axioms:

$$\text{(C3-HQA)} \vdash \text{COMMAND}(\omega_{hq}, \omega_{do}, \phi) \land \text{CONTROL}(\omega_{hq}, \omega_{do}, \phi) \land \text{COMMS}(\omega_{hq}, \omega_{do}, \phi)$$

$$\text{(CM-DA)} \vdash \text{COMMAND}(\omega_{do}, \omega_i, \phi) \text{ where } 1 \leq i \leq n$$

$$\text{(CO-DA)} \vdash \text{COMMS}(\omega_i, \omega_{do}, \phi) \text{ where } 1 \leq i \leq n$$
(CM-DOA) \( \text{COMMAND}(\omega_i, \omega_{do}, \phi) \land \text{SUBTEAM}(\omega_d, \omega_j) \supset \text{COMMAND}(\omega_i, \omega_j, \phi) \)

(CO-DOA) \( \text{COMMAND}(\omega_i, \omega_{do}, \phi) \land \text{SUBTEAM}(\omega_{do}, \omega_j) \supset \text{COMMS}(\omega_j, \omega_i, \phi) \)

Note that if in the headquarters all the sub-teams share the command then if we add transitivity of command between the sub-teams of the headquarters and the divisionalized organization we will get that any sub-team of the headquarters can command any of the divisions. This may or may not be the case depending on the details of the authority of the sub-teams in the headquarters.

It is important to note here that Mintzberg’s description of the divisionalized organization [79] is relatively informal. As such there are different ways to provide a formal model of this organization using the approach described here. Such formalizations may differ in the details of the level of authority and responsibility between the three types of organizations, namely, the headquarters, the division, and the divisionalized organization itself.

We refer to an organization-oriented system with the above axioms as the BGI \(\text{ooos} - \text{do} \) CTL system. Note that the only two axioms introduced here are (CM-DOA) and (CO-DOA). The semantic conditions that correspond to these two new axioms we require that for every organizations \( \omega, \omega' \), and \( \omega'' \) the following conditions hold:

(CM-DOC) \( \forall w \forall t \forall v \text{ if } (\omega_i, \omega_j, w, t, v) \in \mathcal{CM} \text{ then } (\omega_{do}, \omega_j) \in \mathcal{ST} \text{ or } (\omega_i, \omega_{do}, w, t, v) \in \mathcal{CM} \);

(CO-DOC) \( \forall w \forall t \forall v \text{ if } (\omega_j, \omega_i, w, t, v) \in \mathcal{CO} \text{ then } (\omega_{do}, \omega_j) \in \mathcal{ST} \text{ or } (\omega_i, \omega_{do}, w, t, v) \in \mathcal{CM} \).

We refer to an organization oriented model in which the above semantic conditions hold as \( M^{\text{ooos} - \text{do}} \).
Chapter 10

Operational Semantics

In previous chapters we introduced a logical framework for building organization-oriented systems. Such a framework provides the underlying theory for developing distributed systems that exhibit complex behavior. This behavior is specified in terms of the relationship between different sub-systems, i.e., social relationships between the organizations, and the different components of each of these sub-systems, i.e., the joint mental attitudes of each organization. Here we provide a detailed description of how such a specification is translated into an organization-oriented system.

In Chapters 8 and 9 we provided a number of axiom systems that are a specialization of an organization-oriented model. For each such axiom systems the additional axioms specified would be reflected in a specialized operational semantics for the specialized model. Here we provide the operational semantics for one particular organization-oriented system. The axioms that constrain the behavior of this system are the Basic Axioms (see Chapter 8) and the dominant coalition axioms (see Chapter 9).

10.1 Operational Semantics for Agents, Teams, and Organizations

In the development of models of a real-time embedded distributed system we have adopted an incremental approach. We started with a model of an agent-oriented system, through models of multi-agent systems and team-oriented systems to the development of a model of an organization-oriented system. Before we proceed to provide a detailed definition of an operational semantics for an organization-oriented system let us provide a short description of the operational semantics for agent-oriented systems, multi-agent systems, and team-oriented systems.

Recall that we have shown in previous chapters that the model of an agent in a single agent world is a special case of a model of an agent in a multi-agent world. We have then shown that the model of an agent in a multi-agent world is a special case of the model of a team, which in turn is a special case of the model of an organization. Here we show that a similar pattern exists with respect to the architecture and operational semantics of
these models. In particular we show how the architecture and operational semantics of an organization-oriented system is an extension of the architecture and operational semantics for a single agent c.f. Rao and Georgeff [85, 90].

At a high-level the operational semantics of the various systems follow similar principles to the principles described by Rao [85, 90] in his work on a unified theory of plan execution. We also consider that the operation of an agent in a single agent world, agent in a multi-agent world, team, and organization all include three major processes. These are means-end reasoning, deliberation, and reconsideration [85]. Means-end reasoning involves the processing of percepts from the external environment and internal goals and generating possible means for responding to the external percepts or achieving goals.

Given the options generated by the means-end process, deliberation involves deciding on the “best” means that should be employed. Reconsideration involves a process of re-evaluation of the given options in-light of changes to circumstances, failed attempts, or new mental attitudes. Reconsideration occurs prior to deliberation. Deliberation can thus cause reconsidered intentions being ignored.

It is important to note here that although we adopt similar principles in the operational semantics of an agent, team, and organization, there are some major differences between these approaches. In particular the basic executing unit is different. That is, in the team-oriented and organization-oriented approaches the team (and respectively the organization) is actually executing the directives of the specified control-loop. The relationship between the execution process of the team and the execution process of the sub-teams (which are also teams) is defined by the relevant axiom system.

Another difference between the approaches are the means available for achieving goals. In the agent-oriented approach the only means available to an agent are its own plans. In the team-oriented approach the means include the set of sub-teams and the possible allocation of tasks to these sub-teams. That is, the team can modify its set of sub-teams so that the new set has the capability of achieving the goals of the team. In the organization-oriented approach one also considers the way the set of sub-teams are organized in determining goals, making decision, and communicating information. The possible adopted organizational structures and the possible assignment of sub-teams to the different roles are also considered as part of the means available to the organization.

### 10.1.1 Operational Semantics for Agent-Oriented Systems

Recall that the model of an agent operating in a single agent world \(^1\) involves representation of knowledge which only includes the agent or the world. The operational semantics involves the description of how this knowledge is used in the operation of a single agent. The operational semantics is according to the operational semantics described by Rao and Georgeff [85, 90]. We bring it here for completeness of this work.

---

\(^1\)Although there may be multiple agents operating in the system from the agent’s perspective everything outside its own operation is “the world”. There is no distinction between intentional actors causing intending changes to the world and the changes that occur unintentionally.
The architecture of an agent includes declarative knowledge, procedural knowledge, and a control loop that operates the agent. The declarative knowledge is represented by the mental attitudes (i.e., beliefs, goals, and intentions). The procedural knowledge is represented by plans that describe the means of achieving goals under particular circumstances [85].

We refer to the set of goals of an agent as the set of triggering events. At any given time the agent may have multiple events to process, multiple plans that could be used, and multiple intentions that should be executed. We thus also provide to the agent with selection functions for making the above decisions. We thus formally define the architecture for an agent as follows:

**Definition 29**

The architecture of an agent is a tuple

\[ \langle P, A, E, B, I, S_E, S_P, S_I \rangle \]

Where \( P \) is a set of plans, \( A \) is a set of actions, \( E \) is a set of triggering events, \( B \) is a set of beliefs, and \( I \) is thus a set of intentions and is referred to as the intention structure.

The selection function \( S_E \) selects an event from the set \( E \); the selection function \( S_P \) selects an applicable plan from a set of applicable plans; and the selection function \( S_I \) selects an intention from the set of intentions and an activity to perform from the selected intention.

The following main control loop is for the architecture of a single agent (see Figure 10.1). The main objective of the agent is to achieve adopted goals or react to environmental changes. Each such event is referred to as a triggering event. We will start the main control loop with the process of observing the environment. The main control loop includes the following abstract steps:

**Main Control Loop:**

1. Observe the environment. If there are changes to the perceived environment, create a triggering event and add it to the set of triggering events \( E \);
2. Select a triggering event \( e \) using the selection function \( S_E \).
3. Select a Plan \( p \) from \( P \) for responding to \( e \), using the selection function \( S_P \);
4. Form an Intention using \( e \) and \( p \) and Add the Intention to the intention structure \( I \);
5. Select an Intention \( i \) from \( I \);
6. Select an activity \( \alpha \) in \( i \) using the selection function \( S_I \);
7. Execute the Activity \( \alpha \);
10.1. OPERATIONAL SEMANTICS FOR AGENTS, TEAMS, AND ORGANIZATIONS

CHAPTER 10. OPERATIONAL SEMANTICS

Agent in Single-Agent System

Figure 10.1: Control Loop of an Agent in a Single-Agent world.

8. Generate new goal events and add them to the set of triggering events \( E \);

9. Return to Step 1.

10.1.2 Operational Semantics for an Agent in a Multi-Agent Systems

Recall that the model of an agent operating in a multi-agent world involves representation of knowledge which also includes other agents that operate in the world. The operational semantics involves the description of how this knowledge is used in the operation of the agent. The primary differences from agent in a single agent world are:

- the ability to represent the mental attitudes of other agents;
- the ability to identify changes to the environment as a direct result of intentional actions of other agents;

© Copyright Gil Tidhar, 1999
• the ability to deliberate about the mental attitudes and actions of other agents; and
• the ability to communicate with other agents.

Note that the ability to identify changes to the environment as a direct result of intentional actions of other agents is the basis for the work on mental state recognition by Rao and Murray [84, 92]. Further note that the ability to deliberate about the mental attitudes and actions of other agents and the ability to communicate with other agents are the basis for cooperative activity. Such activity is implemented as a combination of independent individual activity.

Again the architecture of the agent includes declarative knowledge, procedural knowledge, and a control loop that operates the agent. The declarative knowledge is represented by the mental attitudes (i.e., beliefs, goals, and intentions). These mental attitudes are with respect to the agent holding them as well as other agents. The procedural knowledge is represented by plans that describe the means of achieving goals under particular circumstances. In particular it may include consideration of the mental attitudes of other agents as well as communicative actions.

We refer to the set of goals of an agent as the set of triggering events. At any given time the agent may have multiple events to process, multiple plans that could be used, and multiple intentions that should be executed. We thus also provide to the agent with selection functions for making the above decisions. We thus formally define the architecture for an agent as follows:

**Definition 30**

The architecture of an agent is a tuple

\[ \langle P, AK, A, E, B, I, S_E, S_P, S_I \rangle \]

Where \( P \) is a set of plans, \( AK \) is knowledge the agent has of other agents in the world, \( A \) is a set of actions (including a set communicative actions), \( E \) is a set of triggering events, \( B \) is a set of beliefs (including beliefs about the actions performed by other agents and the mental state of those agents), and \( I \) is thus a set of intentions and is referred to as the intention structure.

The selection function \( S_E \) selects an event from the set \( E \); the selection function \( S_P \) selects an applicable plan from a set of applicable plans; and the selection function \( S_I \) selects an intention from the set of intentions and an activity to perform from the selected intention.

The following main control loop is for the architecture of a single agent in a multi-agent world (see Figure 10.2). The main objective of the agent is to achieve adopted goals or react to environmental changes. Each such event is referred to as a triggering event. We will start the main control loop with the process of observing the environment. The main control loop includes the following abstract steps:
10.1. OPERATIONAL SEMANTICS FOR AGENTS, TEAMS, AND ORGANIZATIONS

CHAPTER 10. OPERATIONAL SEMANTICS

**Agent in Multi-Agent System**

![Control Loop Diagram]

Figure 10.2: Control Loop of an Agent in a Multi-Agent world.

**Main Control Loop:**

1. Observe the environment (including observation of actions performed by other agents). If there are changes to the perceived environment, create a triggering event and add it to the set of triggering events $E$;

2. Select a triggering event $e$ using the selection function $S_E$.

3. Select a Plan $p$ from $P$ for responding to $e$, using the selection function $S_P$ (taking into account the believed mental states of other agents);

4. Form an Intention using $e$ and $p$ and Add the Intention to the intention structure $I$;

5. Select an Intention $i$ from $I$;

6. Select an activity $\alpha$ in $i$ using the selection function $S_I$;
7. Execute the activity $\alpha$ (including communicative actions);

8. Generate new goal events and add them to the set of triggering events $E$;

9. Return to Step 1.

10.1.3 Operational Semantics for Team-Oriented Systems

The move from an agent operating in multi-agent world to a team represents a conceptual change to the basic architecture. The model of a team operating in a multi-team world involves representation of knowledge which has to be synchronized with the knowledge of other teams, namely its subteams. The operational semantics involves the description of how this knowledge is used in the operation of the team. The primary differences from an agent in a multi-agent world are:

- the ability to describe a structure for a group of teams;
- the ability to represent mental attitudes that are synchronized with mental attitudes of other teams; and
- the ability to describe coordinated activity using joint plans.

Note that the ability to represent mental attitudes that are synchronized with mental attitudes of other teams allows the teams to deliberate about common knowledge for the team [131]. Such common knowledge is the basis for coordinated activity as shown by Halpern and Moses [55]. Further note that the ability to describe coordinated activity using joint plans allows the designer to specify the coordinated activity at an abstract level, that is, the level of the team rather than the level of the individual agent.

Again the architecture of the team includes declarative knowledge, procedural knowledge, and a control loop that operates the agent. The declarative knowledge is represented by the mental attitudes (i.e., mutual beliefs, joint goals, and joint intentions) and knowledge of the team structure, i.e., sub-team relationship. The procedural knowledge is represented by joint plans that describe the means of achieving joint goals under particular circumstances [85]. In particular it may include coordinated activity of multiple sub-teams, i.e., specification of the activity required from the sub-teams.

We refer to the set of joint goals of a team as the set of triggering events. At any given time the team may have multiple events to process, multiple joint plans that could be used, multiple allocation of responsibilities to sub-teams, and multiple intentions that should be executed. We thus also provide to the agent with selection functions for making the above decisions. We thus formally define the architecture for an agent as follows:

**Definition 31**

The *architecture of a team* is a tuple

$$\langle JP, TK, A, E, MB, JI, S_E, S_P, S_T, S_I \rangle$$
Where \( JP \) is a set of joint plans, \( TK \) is a set of sub-team relations, \( A \) is a set of actions, \( E \) is a set of triggering events, \( MB \) is a set of beliefs, and \( JJ \) is thus a set of joint intentions and is referred to as the \textit{joint intention structure}.

The selection function \( S_E \) selects an event from the set \( E \); the selection function \( S_P \) selects an applicable joint plan from a set of applicable plans; the selection function \( S_T \) selects an assignment of sub-teams to the various activities specified in the joint plan; and the selection function \( S_I \) selects an intention from the set of intentions and an activity to perform from the selected intention.

The following main control loop is for the architecture of a single team (see Figure 10.3). The main objective of the team is to achieve adopted joint goals or react to environmental changes. Each such event is referred to as a triggering event. We will start the main control loop with the process of observing the environment. The main control loop includes the following abstract steps:
10.1. OPERATIONAL SEMANTICS FOR AGENTS, TEAMS, AND ORGANIZATIONS

CHAPTER 10. OPERATIONAL SEMANTICS

Main Control Loop:

1. Observe the environment. If there are changes to the perceived environment, create a triggering event and add it to the set of triggering events \( E \);
2. Select a triggering event \( e \) using the selection function \( S_E \);
3. Select a Joint Plan \( p \) from \( JP \) for responding to \( e \), using the selection function \( S_P \);
4. Select an allocation of sub-teams specified in \( TK \) to the activities specified in \( p \) and assign responsibilities;
5. Form a Joint Intention using \( e \), \( p \), and the chosen assignment of responsibilities and add the Joint Intention to the joint intention structure \( JI \);
6. Select a Joint Intention \( i \) from \( JI \);
7. Select an activity \( \alpha \) in \( i \) using the selection function \( S_I \);
8. Execute the Activity \( \alpha \);
9. Generate new joint goal events and add them to the set of triggering events \( E \);
10. Return to Step 1.

There are substantial details that are hidden in each of the above steps. In particular the way joint mental attitudes are synchronized.

10.1.4 Operational Semantics of Organization-Oriented Systems

We start by describing the specification of an organization-oriented system. We will then proceed to describe how such specifications are used in an abstract architecture for such a system. We will show how the axioms of the basic organization-oriented system affect the operational semantics of the system. In Chapter 12 we describe the software design of a software system used for developing different organization-oriented systems.

An organization-oriented system includes two primary components: (1) a specification of declarative, procedural, and social knowledge; and (2) an architecture that uses the specification to control its behavior. In the previous chapters we described a theoretical model for some of the specifications. In particular we focused on social and joint mental attitudes. In Section 10.2 we describe in detail declarative, procedural, and social knowledge.

The axiom system used will primarily impact the architecture of an organization-oriented system. That is, it will determine how the specification of declarative, procedural, and social knowledge is to be used in the operation of the organization-oriented system. The architecture for the basic organization-oriented system is described in Section 10.3.

In Chapter 8 we have shown how a single agent is regarded as a special case of an organization-oriented system. The operational semantics for a single agent is a special case of the operational semantics for the organization-oriented system.
10.2 The Specifications

The approach adopted here to the specification of the behavior of an organization-oriented system is similar to the approach adopted by Rao and Georgeff [90] in the specification of agent-oriented systems. A similar approach has been described by us in previous work on team-oriented systems [69]. According to this approach the behavior of an agent or team is determined by the set of individual or joint plans that are made available to it.

In his work on plans as recipes Rao [85] describes a variety of plan types and their operational semantics. All of these plans are context sensitive procedures. They are the means by which an agent or team can achieve goals, react to situations, or recognize the goals of other agents. Here we describe another type of plan - an organizational plan. An organizational plan is a plan that describes the means by which an organization can achieve joint goals or react to situations.

In this work, in addition to organizational plans, we also view organizations and organizational structures as means for achieving the goals of the system. For each organization, the specification of the behavior of an organization is based on the knowledge available to the organization. Such knowledge includes: (1) possible social relationships; (2) organizational plans; and (3) other organizations in the system. Note that we separate the knowledge about possible social relationships from the knowledge about the known organizations and their team and social structures.

We refer to a set of possible social relationships an organization knows about of as a social structure. We refer to the knowledge of one organization about other organizations as internal organizational model.

10.2.1 Social Structures

Recall that an organizational structure for a particular organization is a combination of a team structure and the set of social relationships between the sub-teams. That is, it is a combination of a team structure and social structure for the organization. Like in the systems approach in Organization Theory we view the organizational structure as the means towards the achievements of the goals of the organization. We now follow the approach of Georgeff and Lansky [48] in developing reasoning systems. That is we view the means towards the achievement of goals as a recipe provided in advance [85].

We thus separate the definition of a social structure from the organization that adopts it. In Chapter 8 we described how a social structure for an organization can be viewed as a collection of structures each associated with a particular state formula. Following this approach we describe for each social structure the goal it is relevant for and the circumstances under which it is to be adopted.

When an organization adopts a new goal or observes a change in its environment, it modifies its set of joint goals or mutual beliefs. The event of adding a joint goal or mutual belief is denoted by the operator +. The event of deleting a joint goal or mutual belief is
10.2. THE SPECIFICATIONS

CHAPTER 10. OPERATIONAL SEMANTICS

denoted by the operator $\neg$. We refer to each of these four events as a *triggering event*.\(^2\)

We define a *social structure* to be: (1) a triggering event describing the purpose of the social structure; (2) a state formula describing circumstances under which the social structure can be adopted; (3) a set of role names; and (4) a set of command, control, and communication relationships between these roles. A role name is an organizational variable. That is, it is a variable to which an organization can be assigned. More formally:

**Definition 32**

A social structure $\sigma$ is a tuple

$$\sigma = \langle ps, c, \rho, s \rangle$$

Where $ps$ is a triggering event describing the purpose of the social structure, $c$ is a state formula describing the context condition, $\rho$ is a set of role names $\{\rho_1, \ldots, \rho_n\}$, and $s$ is a set of social relationships of the form $S(\rho_i, \rho_j, \phi)$, where $1 \leq i, j \leq n$ and $S$ is COMMAND, CONTROL, COMMS.

Note that a social structure is an abstract specification. A library of social structures is simply a set of such structures. Each organization is provided with a library of social structures.

An organization adopts a social structure by assigning the roles in the social structure to its sub-teams. This can be viewed as grounding the social structure. Note that at any given time an organization may have multiple adopted social structures. More formally:

**Definition 33**

We define a *role assignment* $\alpha$ to be a pair $\alpha = \langle \rho, \omega \rangle$ where $\rho$ is a role name and $\omega$ is an organization.

**Definition 34**

Given an organization $\omega$ with sub-teams $\omega_1, \ldots, \omega_n$ and a social structure $\sigma$ with a set of role names $\rho_1, \ldots, \rho_m$, we define an *adopted social structure* for $\omega$ to be a pair $\langle \sigma, \Theta \rangle$ where $\Theta$ is a set of role assignments of the form $\langle \rho_i, \omega_j \rangle$, $1 \leq i \leq n$, $1 \leq j \leq m$.

Given a triggering event $t$ we say that a social structure $\sigma$ is *relevant* to this event if and only if the purpose of the social structure unifies with $t$. Given an organization $\omega$ with set of mutual beliefs $MB$ we say that $\sigma$ is *applicable* to $t$ if and only if it is relevant to $t$, its context condition is a logical consequence of $MB$, and there exists a role assignment for $\omega$ and the set of roles in $\sigma$.

**Example 3**

\(^2\)This is similar to the definition of a purpose or triggering event for joint plans as described by us previously \[69\] and in recent work by Rao \[85\] on plans as recipes.
Recall the simple organizational structure as described in Chapter 9. The simple organizational structure can be typically observed in young and small organizations. Examples of this structure are an automobile dealership with a flamboyant owner, and a middle-sized retail store.

A simple organizational structure has only two prime units: the strategic apex (i.e., CEO) and the operating core (i.e., the production unit). It may also have a secretary performing administrative tasks. The simple organization is identified by a small management that holds both the command and control. That is, the CEO of the organization can determine the goals of the organizations and of each of the members. In C3 terms, the CEO has a command relationship with the whole organization and with every member of the organization.

Furthermore, he or she can (and in many cases will) determine for the whole organization and for each of the members of the organization how they will achieve these goals. This can be expressed as a business plan, work plan, or weekly schedule. In C3 terms the CEO has a control relationship with the organization and each of the members.

Communication links are direct between the members of the members of the organization and the CEO. The CEO will often contact the members directly or will adopt an “open door policy” (i.e., he or she will be contacted by the employees on a regular basis). In C3 terms the CEO and the members have bi-directional communication relationships.

Following is an example of particular social structure that can be used by the simple organization for the production of bolts for a customer. We will refer to this social structure as $\sigma_{\text{bolts}}$.

**Purpose:** $+!(\text{JGOAL}(\langle \text{SO}, \sigma \rangle, \text{produce-bolts}((\text{Customer}, \text{NumOfBolts})))$)

**Context:** $\neg \text{MBEL}(\langle \text{SO}, \sigma \rangle, \text{on-leave}(\text{CEO}))$

**Roles:** $\{\text{CEO}, \text{secretary}, \text{production-unit}\}$

**Social Structure:**

```
{COMMAND(CEO,SO,produce-bolts((Customer,NumOfBolts))),
 CONTROL(CEO,SO,produce-bolts((Customer,NumOfBolts))),
 COMMAND(SO,CEO,schedule-production((NumOfBolts,WorkOrder))),
 CONTROL(SO,CEO,schedule-production((NumOfBolts,WorkOrder))),
 COMMAND(SO,secretary,file(WorkOrder)),
 CONTROL(SO,secretary,file(WorkOrder)),
 COMMAND(SO,secretary,send-invoice((Customer,NumOfBolts))),
 CONTROL(SO,secretary,send-invoice((Customer,NumOfBolts))),
 COMMAND(SO,production-unit,manufacture(WorkOrder)),
 CONTROL(SO,production-unit,manufacture(WorkOrder)),
 COMMS(SO,CEO,produce-bolts((NumOfBolts))),
 COMMS(SO,production-unit,manufacture(WorkOrder)),
 COMMS(production-unit,CEO,manufacture(WorkOrder)),
 COMMS(CEO,secretary,file(WorkOrder)),
 COMMS(CEO,secretary,send-invoice((Customer,NumOfBolts))),
 COMMS(CEO,secretary,send-invoice((Customer,NumOfBolts))))
```
10.2.2 Organizational Plans

An organizational plan describes the activity that an organization should take in order to achieve a goal or respond to a situation. Like single agent or joint execution plans [85] an organizational plan has a purpose. That purpose is either a joint goal to be achieved or a situation to respond to. The plan describes the combination of activities the agent should perform in order to achieve its purpose. Like in the case of social structures a plan is an abstract structure.

The purpose of a plan is different from the purpose of the social structure. It also includes a specification of the social structure that should be adopted when using the plan. In describing the actions to be performed or sub-goals to be achieved one can also specify the social structure that should be adopted by the relevant sub-team. The purpose of a plan is thus a pair of triggering event and social structure.

An organization can primarily perform two types of activities in attempting to achieve the purpose of a plan. It can execute an action or it can adopt a joint goal. Both such activities are performed with by an organization with an adopted social structure. Executing an action $a$ by organization $\omega_r$ using a social structure $\sigma_r$ for an originating organization $\omega_o$ with a social structure $\sigma_o$ is denoted by $\ast(a, \omega_r, \sigma_r, \omega_o, \sigma_o)$. Achieving a joint goal $\phi$ is denoted by $!(\phi, \omega_r, \sigma_r, \omega_o, \sigma_o)$. We refer to each of these two activities as a simple activity. One can now use the simple activities to create a complex activity.

Definition 35

An activity is defined recursively as: (1) any simple activity is an activity; (2) if $s_1$ and $s_2$ are simple activities then $\text{AND}(s_1, s_2)$ and $\text{OR}(s_1, s_2)$ are an activity; and (3) if $a_1$ and $a_2$ are activities then the sequence $a_1; a_2$ is an activity.

In a joint plan when specifying a sub-goal to be achieved one can specify the sub-team that should achieve it. In an organizational plan one can also specify the role that should achieve it. When the plan is used this sub-goal will be achieved by the sub-team that has been assigned that role. As an example, adopting the joint goal:

$\text{JGOAL}((\text{production-unit, } \sigma_{pu}), \text{manufacture-bolts}(300))$

on behalf of the simple organization $\text{SO}$, with the social structure $\sigma_{so}$ is denoted by:

$!(((\text{production-unit, } \sigma_{pu}), \text{manufacture-bolts}(300)), (\text{SO, } \sigma_{so})).$

Although plans are executed by particular organizations they are done so in the context of relationships with other organizations. The purpose of the plan includes more information than the purpose of the social structure. In particular, it includes in addition to the triggering event, the following information: (1) an organization required to execute
the plan; (2) a social structure required to be adopted by the executing organization; (3) an organization that originated the triggering event; and (3) the relationships in the social structure adopted by the originating organization that are relevant to the executing organization.

Given an organization $\omega$ and an adopted social structure $\sigma$, we define the social relationships in $\sigma$ that are relevant to $\omega$ to be the subset of social relationships in $\sigma$ in which $\omega$ is one of the organizations in the relationship. We denote this set of relevant social relationships by $\sigma/\omega$.

We define an organizational plan to be: (1) the purpose of the plan; (2) a state formula describing circumstances under which the plan can be executed; and (3) a combination of activities to be performed, i.e., actions to be executed and joint goals to be achieved. Each activity is labelled with the social structure that should be adopted when performing it. Each organization is provided with a library of organizational plans. More formally:

**Definition 36**

The purpose of a plan, $pp$, is a tuple

$$pp = \langle t, \omega_o, \sigma_o/\omega_r \rangle$$

Where $t$ is a triggering event, $\omega_o$ is the originating organization, and $\sigma_o$ is the social structure of the originating organization.

Note that the triggering event $t$ includes a specification of the organization required to respond to the event, $\omega_r$, and the social structure that should be adopted when responding to the triggering event, $\sigma_r$.

**Definition 37**

An organizational plan $p$ is a tuple

$$p = \langle pp, c, a \rangle$$

Where $pp$ is the purpose of the plan, $c$ is a state formula describing the context condition, and $a$ is an activity to be performed in attempting to achieve the purpose of the plan.

The inclusion of information about the required and originating organizations and social structures allows the designer to specify different plans for roles within an organization. That is, the same organization will use a different plan if it has a different role within the same organization or if it has the same role in a different organization. This corresponds to the model of role behavior described by Simon [107].

For each of the simple activities in a plan the default required and originating organizations and social structures are identical to those specified in the purpose of the plan. This default could be overridden by the designer by specifying other organizations and social structures in the activity.
10.2. THE SPECIFICATIONS  

Chapter 10. OPERATIONAL SEMANTICS

Given a triggering event \( t \) and an associated required organization \( \omega_r \), required social structure \( \sigma_r \), originating organization \( \omega_o \), and originating social structure \( \sigma_o \) we say that an organizational plan \( p \) is relevant to this event if and only if each of the components of the purpose \( p \) unifies with \( t, \omega_r, \sigma_r, \omega_o, \) and \( \sigma_o \) respectively. Given an organization \( \omega \) with an internal organizational model \( SK \) and a set of mutual beliefs \( MB \), we say that \( p \) is applicable to \( t \) if and only if it is relevant to it and its context condition is a logical consequence of \( MB \).

Example 4

In Chapter 8 we described the simple organization \( S_0 \). Let us now provide an example of an organizational plan that could be used by \( S_0 \). Let us consider the roles in a social structure for \( S_0 \) to be: CEO, production-unit, and secretary. The organizational plan describes how the simple organization \( S_0 \) achieves the joint goal to produce bolts. We will refer to this plan as \( pp_{bolts} \).

\[
\text{Purpose: } +!(\text{JGOAL}(\langle \omega_r, \sigma_r \rangle, \text{produce-bolts}(\text{Customer,NumOfBolts})), \langle \omega_o, \sigma_o \rangle)
\]

\[
\text{Context: } \neg \text{MBEL}(\langle \omega_r, \sigma_r \rangle, \text{holiday})
\]

\[
\text{Activity: } !(((\text{CEO}, \text{CEO}), \text{schedule-production}(\text{NumOfBolts,WorkOrder}), \langle \omega_r, \sigma_r \rangle))
\]

\[
\text{AND (}
\]

\[
!(((\text{secretary}, \text{sec}), \text{file}(\text{WorkOrder}), \langle \omega_r, \sigma_r \rangle));
\]

\[
!(((\text{production-unit}, \text{pu}), \text{manufacture}(\text{WorkOrder}), \langle \omega_r, \sigma_r \rangle))
\]

\[
);
\]

\[
!(((\text{secretary}, \text{sec}), \text{send-invoice}(\text{Customer,NumOfBolts}), \langle \omega_r, \sigma_r \rangle)).
\]

10.2.3 Internal Organizational Model

Recall that organizations operate in an environment in which there are other organizations. Their ability to operate in this environment will depend on the knowledge they have about these organization. We refer to this knowledge as the internal organizational model.

There are three types of knowledge about an organization: (1) knowledge of the existence of an organization; (2) knowledge of its team structure, i.e., the sub-team relationships it has with other organizations; and (3) knowledge of its adopted social structures, i.e., the social relationships that exist between the sub-teams.

The minimal internal organizational model that an organization has is the knowledge about itself. That is, its existence, sub-teams, and adopted social structures. All the organization’s internal organizational model is contained in a special knowledge base.

Example 5

Let us consider a particular simple organization \( M-Bolts \), where John F. Tie is the CEO, Emily P. Fax is the secretary, and Blue M. Collar is the production-unit.

Let us consider the social knowledge base of the CEO in the simple organization, \( M-Bolts \). The basic social knowledge is the knowledge about the existence and command,
10.3. AN ARCHITECTURE FOR ORGANIZATIONS AND ORGANIZATIONAL BEHAVIOR

CHAPTER 10. OPERATIONAL SEMANTICS

close, and communication relationships of John F. Tie with itself. In addition the CEO is aware of the existence of M-Bolts, its team structure, and adopted social structure. The team structure for M-Bolts is represented by the following set of sub-team relationships:

\{ \text{SUBTEAM}(M-Bolts, John F. Tie), \\
\text{SUBTEAM}(M-Bolts, Blue M. Collar) \\
\text{SUBTEAM}(M-Bolts, Emily P. Fax) \}\}

The adopted organization for M-Bolts is based on the social structure $\sigma_{bolts}$ described in a previous example. The role assignment for this social structure is as follows:

\{ \langle \text{CEO}, John F. Tie \rangle, \\
\langle \text{secretary}, Emily P. Fax \rangle, \\
\langle \text{production-unit}, Blue M. Collar \rangle \}\}

10.3 An Architecture for Organizations and Organizational Behavior

In this section we will describe the operational semantics of an organization-oriented system. In particular we will describe the various processes that take place as part of the architecture of an organization and how these processes are affected by the various types of knowledge described in the previous section. Let us first provide an informal description of the architecture and organizational behavior.

10.3.1 The Components

In the previous section we described the social structure and organizational plan libraries as well as the internal organizational model that are available to an organization. In addition to this knowledge an organization also has a set of triggering events, an intention structure, and a specification of the actions it can perform.

At any given time the organization may have multiple events to process, multiple social structures that can be adopted, multiple organizational plans that could be used, and multiple intentions that should be executed. We thus also provide to the organization with selection functions for making the above decisions. We thus formally define the architecture for an organization as follows:

Definition 38

The architecture of an organization is a tuple

$$\langle SS, P, SK, A, E, MB, I, JI, SE, SO, S_S, SP, SR, SS_I, SI \rangle$$
10.3. AN ARCHITECTURE FOR ORGANIZATIONS AND ORGANIZATIONAL BEHAVIOR

CHAPTER 10. OPERATIONAL SEMANTICS

Where $SS$ is a set of social structures, $OP$ is a set of organizational plans, $SK$ is an internal organizational model, $A$ is a set of actions, $E$ is a set of triggering events, $MB$ is a set of mutual beliefs. We associate with each organization for which the executing organization is a member of a set of intentions. $JI$ is thus a set of sets of joint intentions and is referred to as the joint intention structure.

The selection function $SE$ selects an event from the set $E$; the selection function $SO$ selects an organization from the set of known organizations; the selection function $SS$ selects an applicable social structure from the set of applicable social structures; the selection function $SP$ selects an applicable plan from a set of applicable plans; the selection function $SR$ selects an assignment of roles to the activities in a joint plan; the selection function $SSI$ selects the set of intentions for an organization from $JI$; and the selection function $SI$ selects a joint intention from the set of intentions for a known organization and an activity to perform from the selected intention.

Joint Intention Structure

In the previous sections we distinguished between the executing organization and the originating organization. That is, we distinguished between the organization that responds to a triggering event and the organization that has generated the triggering event. We thus allow for an organization to respond to triggering events generated by multiple organizations.

An organization responds to a triggering event by forming a joint intention to respond to it and then executing this intention. One would thus like maintain the distinguishing information in the way joint intentions are managed. The way this information is used is described as part of the process of selecting a set of intentions from the joint intention structure.

The joint intention structure is composed of a set of joint intentions one for each originating organization with a particular originating social structure. Each set of joint intentions is similar to the set of intentions for a single agent as described by Rao [85].

The adoption of sub-goals causes the creation of a sub-intention. We associate that sub-intention with the sub-goal activity that has caused it. Each intention in the set of intentions may thus include a tree of sub-intentions which are currently being executed. One can thus view an intention as an execution stack of a process in an operating system of a computer. Each sub-intention corresponds to a separate procedure call made by the previous procedure.

10.3.2 Informal Description

In the previous sections we described the specification of a social structure, organizational plan, and internal organizational model. These specifications are used to determine the behavior of an organization. The behavior of the organization is influenced by the social
mental attitudes and its knowledge of other organizations in its environment. We refer to a behavior that is influenced by social mental attitudes as social behavior.

The architecture of an organization that exhibits social behavior is an adaptation of the BDI architecture [29, 89]. In particular an organization includes a mutual belief knowledgebase (MB), a set of joint goals (JG), and a set of joint intentions (JI). In addition the architecture includes a set of known social structures (referred to as a structure library and denoted by SS), a set of organizational plans (referred to as a plan library and denoted by OP), and an internal organizational model (referred to as a social knowledge base and denoted by SK).

In addition to the above knowledge and mental state we include a number of selection functions that reflect the choice available to the organization. That is, they implement the decision procedures used when selecting from different alternative joint goals, social structures, organizational plans, and joint intentions.

Recall that we are interested in software architectures for reactive software systems that are embedded in a dynamic environment. The main process of a single organization can be described as a continuous loop in which multiple sub-processes are called upon. We refer to this as the main control loop.

The following main control loop is for the architecture of a single organization (see Figure 10.4). When discussing the details of this architecture we refer to the organization that has been implemented using the particular architecture as the executing organization.

The main objective of the organization is to achieve an adopted joint goals or react to environmental changes. Each such event is referred to as a triggering event. We will start the main control loop with the process of observing the environment. The main control loop includes the following abstract steps:

Main Control Loop:

1. Observe the environment. If there are changes to the perceived environment, create a triggering event and add it to the set of triggering events JG;

2. Select a triggering event $e$ using the selection function $S_E$.

3. If $e$ does not have an associated organization, Select an Organization $\omega$ to achieve $e$ from the set of known organizations contained in the social knowledgebase $SK$ using the selection function $S_O$;

4. If $\omega$ is not the executing organization Delegate the Responsibility to $\omega$ and goto 1;

5. Select a Social Structure $\sigma$, to be adopted by $\omega$ when responding to $e$, using the selection function $S_S$;

6. Select an Organizational Plan $p$ from $OP$, to be used by $\omega$ with social structure $\sigma$ when responding to $e$, using the selection function $S_P$;
7. Adopt the social structure $\sigma$, select an allocation of sub-teams specified in $TK$ to the activities specified in $p$, and assign the responsibilities;

8. Form a Joint Intention for $\omega$, using $e$, $\sigma$, $p$, and the chosen assignment of responsibilities and Add the Joint Intention to the joint intention structure $JI$;

9. Select a Set of Joint Intentions $I$ from $JI$ using the selection function $S_{SI}$;

10. Select a Joint Intention $i$ from $I$;

11. Select an activity $\alpha$ in $i$ using the selection function $S_I$;

12. Execute the Activity $\alpha$;

13. Generate new joint goal events and add them to the set of triggering events $JG$;


Figure 10.4: Control Loop of an Organization.
There are substantial details that are hidden in each of the above steps. In particular, the way the social relationship with the selected organization $\omega$ affects the above decisions and behavior. Furthermore, the way the selected social structure $\sigma$ affects the way selections are being made. In the following sections, we explore the details of each of these steps.

In general, the social mental attitudes affect the behavior of the organization in the following ways:

- an organization will accept a joint goal only from organizations that command it;
- an organization will communicate to its controlling organizations the possible choices available to it;
- the selection of social structure, organizational plan, joint intention structure, and joint intention, are all done by the controlling organizations;
- an organization will communicate the successful or failed execution of a joint intention to its controlling organizations;
- an organization will communicate the successful or failed attempt to achieve a joint goal to its commanding organizations.
- an organization will communicate changes to its mutual beliefs to the organizations it has a relevant communication relationships with.

**Responding to Failure**

In the previous sections, we described how an activity is performed and how joint intentions are executed. Given that the organization is operating in an uncertain environment, failure can occur at a number of stages. The behavior of an organization is based on a number of decisions that lead to a particular option being selected. These decisions are made by the controlling organizations. The principle in responding to failure is to mark the option as failed, return to the last decision point, and seek an alternative option that has not been marked as failed.

The first possible failure is failure to select an organization that will achieve the joint goal. In this case, the joint goal will be marked as failed. If it has been generated by performing a sub-goal activity, then the sub-goal activity will be marked as failed and an alternate activity will be selected. Otherwise, the joint goal is simply discarded.

The next possible failure is failure to select a social structure for the joint goal and selected organization. In this case, the organization will be marked as failed, and an alternative organization will be selected.

The next possible failure is failure to select an organizational plan for the joint goal, organization, and social structure. In this case, the social structure will be marked as failed, and an alternative social structure will be selected.
The next possible failure is either a failure to create a joint intention for the organization or the failure of the execution of the selected joint intention. In this case the organizational plan will be marked as failed and an alternative plan will be selected.

The next possible failure is failure to perform an activity. If this activity is part of an OR activity then one would simply select the next activity in the OR. Otherwise the overall activity will fail possibly leading to the failure of the joint intention.

Recall the dominant coalition axioms that enforce that the joint intention for an organization be held by its commanding organizations. It follows that any modification to the joint intention or failure of an activity of a joint intention has to be synchronized between the commanding organizations.

Example 6

Let us now describe the situation in which John F. Tie, the CEO of the small manufacturing company M-Bolts, has a joint goal to produce 300 bolts for its loyal customer B. Construction. The process will commence with John F. Tie adding the event +!(JGOAL(ωr, σr, produce-bolts(Customer,NumOfBolts)), ωo, σo). Given that there is no organization associated with this event John F. Tie will select an organization that can achieve the joint goal. An obvious candidate is M-Bolts.

The joint goal will thus be delegated to M-Bolts. Given the command relationship M-Bolts has with John F. Tie it will adopt the joint goal. It will now have to select an appropriate social structure to be adopted for the purpose of achieving this joint goal. This selection will be made by the controller of M-Bolts, that is, John F. Tie. A possible selection would be the already adopted social structure σbolts.

At this stage an organizational plan has to be selected. Again the selection will be made by John F. Tie. Let us assume that the plan selected would be ppbolts.

The plan ppbolts would now be used to form a joint intention. Given that John F. Tie is the organization controlling M-Bolts then the adoption of the new joint intention has to be synchronized with it.

At this stage a joint intention has to be selected. If there are multiple joint intentions then John F. Tie will again have to make a decision as to this choice. After the intention has been selected it will be executed.

The first activity is delegating the scheduling of the production to John F. Tie. The second activity involves delegating the filing of the work order to Emily P. Fax and the manufacturing of the bolts to Blue M. Collar.

Note that in the plan for producing the bolts M-Bolts would be the required organization and John F. Tie would be the originating organization. In the manufacturing of the bolts Blue M. Collar would be the required organization and M-Bolts would be the originating organization.

Given that M-Bolts controls Blue M. Collar it should make the decisions for it. But, given that M-Bolts is in turn controlled by John F. Tie then actually it will make all the control decisions for Blue M. Collar. Similar consideration apply to the filing of the work order and sending the invoice achieved by Emily P. Fax.
10.3.3 The Processes

Adding a Triggering Event

Triggering events are generated by changes to the mutual beliefs or the adopted goals of an organization. Triggering events may have an organization and social structure associated with them.

When the organization adds or deletes a mutual belief about the environment a new trigerring event is created. This event is added to the set of events $E$. Similarly when an organization adopts or discards a joint goal a triggering event is generated. Again this event is added to $E$.

A Joint goal $g$ is adopted by performing the simple activity $!(\omega, \sigma, g)$. Note that this activity allows the designer of the system to specify the organization $\omega$ and social structure $\sigma$ that should be associated with the new triggering event. This specification will both guide and constrain the behavior of the system when considering the possible means of responding to the triggering event.

In our architecture performing the activity of adopting a joint goal will occur in the context of a joint intention. Such joint intentions are executed by an organization and with an adopted social structure. This Information about the originating intention is also associated with the triggering event.

Selecting an Organization

If an organization has already been specified for the triggering event then the selection process simply returns the specified organization. Alternatively an organization that will be best suited to respond to the triggering event has to be selected.

There are two problems that need to be addressed: (1) which organization will make the selection; and (2) how to make the selection. The selection of an organization depends on the skills and capabilities of that organization. In Chapter 11 we provide an approach to the guided selection of organizations. Here we will address the problem of determining which organization will make the selection.

Recall that a control relationship implies that the controlling organization determines the means by which a goal is to be achieved. In the context of the architecture this implies that a controlling organization will make decisions for the controlled organizations. In particular a controlling organization will select an organization that will respond to the triggering event.

The triggering event may have been generated from a joint intention. In this case it will have associated with it the original organization and social structure under which the event has been created. This social structure specifies which organizations control the executing organization. All of these controlling organizations are responsible for jointly making the selection for the executing organization.
Alternatively the event may have been generated because of a change in the mutual beliefs of the organization. Such a triggering event does not have an originating organization and social structure associated with it. In this case it is the responsibility of the executing organization to select the responding organization.

The executing organization has to convey to the controlling organization the need for a decision to be made. This may involve a sequence of communications at the end of which the executing organization will be informed of the decision, i.e., the selected organization.

**Delegating a Responsibility**

After an organization has been selected the executing organization has to delegate the responsibility of responding to the triggering event to that organization. Such delegation can only occur if there is a command relationship between the executing organization and the selected organization. If such a relationship does not exist then another organization has to be selected.

It is important to note here that an organization has been selected by the controlling organizations. The internal organizational model of the controlling organizations may be different than the internal organizational model of the executing organization. This may result in a potential conflict as would be the case in human organizations.

One way to eliminate this potential conflict is to communicate to the controlling organizations the information associated with the triggering event. That is the originating organization and the adopted social structure of the originating organization. This information will include a set of social relationships between the executing organization and other organizations. Such communication corresponds to the staff information flow described in Chapter 8.

If the command relationship between the executing organization and the selected organization does exist then the process of delegation will commence. Delegation of a responsibility involves three distinct steps: (1) the transfer of responsibility; (2) acting on this responsibility; and (3) communicating the result of the actions, i.e., success or failure.

The first step is the process of one organization exercising its authority over another organization. It reflects the basic organization-oriented systems, that is, the relation between the command relationship and the joint goals of the two organizations. The third step corresponds to the performance information flow described in Chapter 8.

**Selecting a Social Structure**

The process of selecting a social structure is similar to the process of selecting an organization. Again there are two problems that need to be addressed: (1) who will select the social structure; and (2) how is a social structure selected.

If the triggering event has a required social structure associated with it then this structure will be selected. Alternatively the controlling organizations are responsible for selecting a social structure.
10.3. AN ARCHITECTURE FOR ORGANIZATIONS AND ORGANIZATIONAL BEHAVIOR

CHAPTER 10. OPERATIONAL SEMANTICS

The selection of a social structure is based on the information contained in the library of social structures and the internal organizational model. Based on this library one can generate a set of applicable social structures for the triggering event. One can then generate a role assignment for each of the applicable social structures. The selection function \( S \) is then used to make the selection. Note that the selection also includes the role assignment for the social structure.

It is important to note that if the controlling organization is making the selection then the evaluation of applicability will be based on the set of mutual beliefs \( MB \) of the controlling organization and not the executing organization. Again this may lead to a potential conflict as the selected social structure may not be applicable when evaluated by the executing organization. One way to eliminate this potential conflict is to also communicate to the controlling organizations the set of applicable social structures as evaluated by the executing organization. Such communication corresponds to the staff information flow described in Chapter 8.

Selecting an Organizational Plan

The process of selecting an organizational plan is similar to the process of selecting a social structure. Again we would have the controlling organizations making the decision on the choice of applicable organizational plan. Details of how to form the set of applicable plans can be found elsewhere [85].

The primary difference between the selection of a social structure and an organizational plan comes from the difference in the purpose of the two. That is, selecting an organizational plan also involves the unification between the required and originating organizations and social structures.

Adopting a Social Structure

After all the decisions as to the means of responding to a triggering event have been made the organization is now required to act on these decisions. Acting on these decisions involves adopting the social structure, creating a joint intention, and adding the new joint intention to the set of intentions for the relevant organization.

The process of an organization \( \omega \) adopting a social structure \( \sigma \) with an associated role assignment \( \alpha \) involves two steps: (1) adding \( \sigma \) and \( \alpha \) to the knowledge about \( \omega \) contained in the internal organizational model; and (2) synchronizing the organizational model of \( \omega \) with other organization.

The first step is simple and will not be discussed here. As to the second step, recall that we include the dominant coalition axioms in this model of organization-oriented systems. According to these axioms only the commanding organizations determine the mental attitudes of the organization.

A social structure is adopted as the means for responding to a triggering event. This event may also have associated with it the information on the originating organization and its adopted social structure. We thus require that when an organization adopts a social
structure the internal organizational model of the commanding organizations according to the originating social structure be updated. If the triggering event does not have an originating social structure then there is not required synchronization.

**Forming a Joint Intention**

Given the organizational plan, required organization, originating organization, and originating social structure the organization is now in a position to create a joint intention. As described by Rao [85] creating an intention is a process of instantiating the plan with the associated unifying information. This process is also true for a joint intention and an organizational plan.

Recall that an organizational plan can also have role names in the activities it describes. The role assignment can thus be viewed as another unifier. The process of instantiating an organizational plan also involves instantiating the role names with the assigned sub-teams.

The main difference between an intention and a joint intention is that one also has to synchronize the creation of the joint intention with the dominant coalition. Note though that the joint intention does not impact the behavior of the organization until it is added to the set of joint intentions for the organization. We thus leave the synchronization process to the next step in the control loop.

**Adding a Joint Intention to Joint Intention Structure**

A joint intention for an organization can be created because of one of three situations: (1) this is a joint intention created to achieve a sub-goal generated by the same organization as part of executing another joint intention; (2) this is a joint intention created to achieve a sub-goal generated by another organization as part of executing another joint intention; or (3) this is a joint intention created as a response to a triggering event with no originating organization.

We follow the terminology of Rao [85] and refer to the new joint intention created in the first situation as a sub-joint-intention. We refer to joint intentions in the other two situations as a root-joint-intention.

Both types of joint intentions will be added to the set of joint intentions for the relevant organization. The process of adding a joint intention to the set of joint intentions will depend on the type of joint intention. A sub-joint-intention is added to an existing joint intention. A root-joint-intention is added as a new element of the set of joint intentions.

As in the case of adopting a social structure, we are also required to synchronize the set of joint intentions for the organization with the dominant coalition of the organization. In this case we need to ensure that all of the controlling organizations in the adopted social structure also adopt the corresponding joint intention.

**Selecting a Set of Joint Intentions**

The executing organization is now in a position to select a joint intention to execute. The joint intention structure of the executing organization includes multiple sets of joint
intentions adopted by the organization on behalf of other organizations. The executing organization now has to express its preference as to its commitment to different organizations.

The decision may depend on its role within the social structure adopted by these organizations, previous relationships, or some utility function (e.g., one organization pays it more for its services than the other). We refer to this decision as a social preference. A detailed description of social preference is provided by Cavedon and Sonenberg [16]. Here we include the social preference in the selection function $S_{SI}$.

Selecting a Joint Intention

After a selection has been made as to the organization for which the next intention is to be executed one has to select the next joint intention from the set of intentions. Again this decision is to be made by the controllers of the organization. Again there may be conflict between the executing organization and the controlling organizations as to the options available.

One can overcome this conflict by communicating the set of joint intentions to the controlling organizations. Such communication corresponds to the staff information flow described in Chapter 8.

It is important to note here that a joint intention is derived from an organizational plan and as such is hierarchical in nature (i.e., it includes a specification of sub-goals). In the selection of a joint intention there is thus the implicit command to adopt the sub-goals when required. The controlling organization is thus required to have a command relationship with respect to the sub-goals specified in the joint intention. This requirement imposes constraints on the joint intentions that a controlling organization can select for the controlled organization.

An example of such a situation can be clearly identified in a military unit. Given a command to perform an attack an established plan of attack may include an illegal command, that is, a command that is outside the bounds of authority. It is thus the responsibility of the commanded organization to reject this command and refuse to execute that part of the established plan of attack.

Selecting an Activity

In addition to the selection of a joint intention one has to select the activity to be performed. If there are options then the selection is made by the controlling organizations and communicated back to the executing organization.

The activities in an intention are in one of the forms $AND(a_1, a_2)$, $OR(a_1, a_2)$, or $a_1; a_2$ (i.e., a sequence). If the activity is in the form $AND(a_1, a_2)$ then one has to perform both $a_1$ and $a_2$. The order of execution is not determined. The success of an $AND$ activity will occur if both $a_1$ and $a_2$ are marked with success. We then mark the $AND$ activity as success, otherwise we mark it as failed.
If the activity is in the form \( OR(a_1, a_2) \) then one has to perform at least one of \( a_1 \) and \( a_2 \). The order of execution is not determined. The success of an \( OR \) activity will occur if one of \( a_1 \) or \( a_2 \) are marked as success. We then mark the \( OR \) activity as success, otherwise we mark it as failed.

If the activity is in the form \( a_1; a_2 \) then one has to perform first \( a_1 \) and then \( a_2 \). The success of a sequence activity will occur if both \( a_1 \) and \( a_2 \) are marked as success. We then mark the sequence activity as success, otherwise we mark it as failed.

Using the above definition of success one can now evaluate the success of a joint intention by recursively evaluating the activities. This is done in a bottom-up approach. The evaluation of success or failure of simple activities is done by performing them. This is described in the following section.

### Executing a Joint Intention

The process of executing a joint intention is identical to the process of executing a single agent intention [85]. That is, it involves performing the selected activity. It may be the case that there are no more activities to perform. This can either be because the intention has completed successfully or because it has failed. In either case the joint intention will be marked as succeeded or failed. In the following section we describe how failure is handled.

As mentioned performing an activity may involve executing an action or adopting a new sub-goal. Executing an action may either succeed or fail depending on the action and the state of the environment. Adopting a sub-goal involves adding a new triggering event to the set of events \( E \). The sub-goal is successful if the attempt to respond to the relevant triggering event is successful. In either case the activity will be marked with the result of the execution. This marking will be taken into consideration in the selection of the next activity to be performed.

The attempt to respond to a triggering event is successful if there is a corresponding joint intention that has been marked as success. Note that that joint intention may have caused other joint intentions to be created. That is, the execution of a joint intention may generate a tree of sub-intentions.

### 10.4 Coordinating Multiple Organizations

In the previous sections we described the architecture and operational semantics for an organization-oriented system. In many of the steps of the main control loop we have indicated the interactions between multiple organizations. We have shown how the social mental attitudes affect these interactions. Here we provide details of the way the activities of an organization’s sub-teams are coordinated.

All the activities of an organization are specified in the organizational plans stored in the organizations plan library. Recall that the activity in an organizational plan specifies the components of this activity in terms of simple, \( AND \), \( OR \), and sequence activities. Each simple activity is labelled by the organization (or sub-team) that is responsible to
perform this activity. Furthermore the plan also describes the temporal order in which each of the activities is to be performed.

The organizational plan is thus a recipe for the coordinated activity of the sub-teams. This is similar to the approach described by us previously [69] and further elaborated by Tambe [117, 119] in the implementation of joint plans for team activity. The coordination specified in an organizational plan is used when the plan is instantiated to form a joint intention. The coordinated activity of an organization is thus represented by the joint intention it is executing.

Recall that based on the dominant coalition axioms an organization has a joint intention only if all its controllers have this joint intention. This is ensured in the process of synchronizing the joint intention structure taking place when a joint intention is adopted. The other sub-teams of the organization are not required to adopt the joint intention for the organization.

It follows that actually only the controlling organizations need to execute the joint intention. All other sub-teams of the organization are only required to execute their part of the joint intention. This is done through the process of delegation as described above.

It is important to make a number of comments on the issue of performance. First, when there are multiple controlling organizations then they all must execute the joint intention. This is achieved by continuously synchronizing the state of the execution of the joint intention between them. This synchronization overhead can be reduced if there is only a single controller for an organization as is the case in a military unit.

Second, in his work on joint plans Tambe [117] has demonstrated that the use of predefined joint plans improves the coordinated activity of a group of artificial agents. This is achieved by making the joint intentions common knowledge for the members of the group. Here we make the joint intentions common to the dominant coalition and add communication between the dominant coalition and the other sub-teams.

Recall that in Chapter 7 we described one particular team-oriented system as a generalization of the Halpern and Moses [54, 55] approach. Halpern and Moses show that true coordination is only possible in the presence of common knowledge. They proceed to show a number of approximations of common knowledge. Indeed one can consider our model of organizational plans in the context of the work of Halpern and Moses on common knowledge [55]. One can view joint plans as another type of approximation of common knowledge used for coordination.

10.5 Evaluation of the Organization-Oriented Approach

In the previous chapters we have described in detail a theoretical model and the operational semantics of organization-oriented systems. In this chapter we provide a short analysis and evaluation of the organization-oriented approach. We will identify some of the advantages and disadvantages of adopting such an approach. It is important to note that the objective of the work described in this section is not to provide a complete analysis and evaluation. Rather it is intended to highlight some of the advantages and problems of the organization-
10.5. EVALUATION OF THE ORGANIZATION-ORIENTED APPROACH

We also provide some indication of further evaluation that could be performed.

The analysis presented in this section of the organization-oriented approach will be along two dimensions. The first will involve an analysis based on the usability of the models and systems presented. This analysis will highlight the ease and difficulty in modelling particular types of multi-agent systems. The second analysis will involve an analysis of the complexities that are introduced in the organization-oriented model, e.g., dependencies between specifications, the need for explicit specifications, etc.

10.5.1 Usability

It is important to note that the organization-oriented approach is primarily designed to be a software engineering model that allows designers to build distributed systems. We believe that the resulting implemented system could have been built using an agent-oriented approach, object-oriented approach, or a structured approach. The issue is not the ability to implement a system but rather how easy it is to do so using a particular model.

One way of analysing such a software engineering model is by analysing the ease (or difficulty) in which one can design a distributed systems using it. The ease of use will be determined by two factors: (1) the ease of specifying a particular distributed system; and (2) the ease in which this specification is then transformed into an implemented system.

Ease of Specifying a Distributed System

An organization-oriented system is primarily composed of specification of organizations, organizational structures, organizational plans, internal organizational model, mutual belief knowledgebase, and an execution model that operates based on these specifications.

In general we argue that the organization-oriented approach provides a number of benefits: (1) it allows the designer to reason explicitly about the various sub-systems of a distributed system at an abstract level; (2) it allows the designer to explicitly specify the relationships between the behavior of a system and the behavior of the sub-system; (3) it allows the designer to specify how the various sub-systems coordinate their behavior; and (4) it allows the designer to specify how the various coordination mechanism can be modified dynamically in a context sensitive way.

In the same token we argue that the organization-oriented approach has a number of limitations: (1) the formal specification of the distributed system will include elements which may not explicitly exist in other approaches but have to be formally specified here, e.g., coordination mechanisms, sub-systems, relationships between sub-systems and the whole system, etc.; (2) the social relationships between the sub-systems are limited to command, control, and communication; and (3) the model does not consider limitations on underlying communication links which are the basis for the interaction between the sub-systems.

The view of a system or a part of a system as an atomic unit exists in any modular design of a software system. It is true that this view may be documented only in the design
description, as a conceptual sub-system in the architectural design, or in the designers head. Nevertheless this view does exist. In the organization-oriented approach we simply allow for this view to be expressed explicitly rather than implicitly.

A distributed system may have multiple sub-systems that operate simultaneously. If these sub-systems are required to coordinated their activities (due to shared resources, task dependencies, etc.) then a coordination mechanism is required. These coordination mechanisms would specify the order in which various tasks are to be performed and the sub-system that will be responsible for executing each task.

These coordination mechanisms will then be translated into specific behaviors that will be independently executed by each of the sub-systems. The combination of these behaviors will represent a coordinated activity of the system as a whole. The organization-oriented approach offers a mechanism for specifying the coordinated activity of the system in the form of an organizational plan. This specification is then automatically used to direct the behavior of the sub-systems.

The coordinated behavior may also requires the sharing of information or joint decision making as to multiple options. The designer will be required to design which sub-systems share information and how the various sub-system reach joint agreement. Again this design will be translated into the specific behaviors that will be independently executed by each of the sub-systems. The organization-oriented approach offers a mechanism for specifying the relationships between the sub-systems in the form of an organizational structure. This specification is then automatically used to direct the behavior of the sub-systems.

Similar arguments hold for the internal organizational model and the mutual belief knowledgebase. An execution engine that implements a particular operational semantics is the basis for any running system.

One important aspect of an organization-oriented model is the flexibility in changing the coordination mechanism. In particular the model allows for dynamic re-configuration of the various components into systems and sub-systems. Furthermore the coordination mechanisms used by the system and sub-systems can be changed dynamically.

These re-configurations can be done in a context sensitive way. The system can thus change its configuration based on the parameters such as load, abilities, preferences, and authorizations of the various sub-systems.

This additional flexibility comes directly from the fact that the sub-systems and coordination mechanisms are represented explicitly as first class entities in the model.

**From a Specification to an Implementation**

The ease of transforming a specification of an organization-oriented system into an implemented system will depend on the automated tools available. In Chapter 12 we have described the design of a system that takes organization-oriented specifications and uses them to generate an implemented system. As mentioned above this is a system for generating organization-oriented systems. This design has been implemented as prototype system.
From a software engineering perspective the above design and prototype implementation is a type of 4GL implementation. We argue that given this 4GL, from the designers point of view, the transformation from a specification of an organization-oriented system to an implemented system is trivial. Like other 4GL’s it amounts to issuing the command to automatically generate the implemented system and then issuing the command to run this system.

It is important to note here that the design and implementation are only for the basic organization-oriented model. Furthermore, these are only in prototype form and the system has not been used in developing commercial applications.

10.5.2 Complexity

In any distributed system there a variety of dependencies between the choice of sub-systems, coordination mechanisms used, and the information that is stored in each sub-system as to its own state and the state of other sub-systems. These dependencies can also be observed in organization-oriented systems.

Given that a specification of an organization-oriented system includes an explicit model of the sub-systems and their interactions the dependencies between the specifications are also made more explicit. It is important to note that the dependencies between the organizations, organizational structures, organizational plans, internal organizational model, and mutual belief knowledgebase can be very subtle. Let us now describe some of these dependencies. We refer to some of them in the analysis and design methodology described in the following sections.

As an example consider an organizational plan that could be adopted by an organization. This plan may specify a variety of roles and an allocation of responsibilities to these roles. The plan thus depends on the roles specified in the organizational structure adopted. Furthermore the ability to execute an organizational plan will now depend on the combination of abilities of the various organizations that have been assigned to the specified roles (see Chapter 11 for a description of organization selection).

The choice of organizational structure that can be adopted may depend on the organizations available to adopt such a structure. That is, for an organization to adopt a structure it needs to both have the structure in its library as well as have a valid context for the structure (i.e., for an organizational structure to be adopted it has to be both relevant and applicable - see Chapter 10 for further details).

Even if the organizational structure has been adopted, different organizations may have different models of the organization. This is based on a different internal organizational model. The behavior of the organizations will also depend on their internal organizational model. It follows that there is a potential for problems in the coordination of the behaviors if there are inconsistencies between the internal organizational models of different organizations.

It seems then that the specification of a distributed system is very complex. The organization-oriented approach does not eliminate this complexity. Rather, it allows the designer of a system to explicitly reason about this complexity.
10.5.3 Further Evaluation

Evaluation of a software development methodology is by no means a simple task. Wood et. al. [150] suggest that an evaluation of a software development method would include:

**Need Analysis:** Determining the important characteristics of the system to be developed and how the method helps developers deal with those characteristics.

**Constraint Identification:** Identifying the constraints imposed on the permitted solutions and determining how the method helps developers deal with those constraints. A method must support developers’ efforts to design systems that exhibit a variety of required characteristics.

**User Requirements:** Determining the general usage characteristics of a method. A method can be examined by developing an understanding of: how it represents a system under development, the guidelines it gives developers to derive the representations, and the guidelines it provides to examine the representations. This understanding is best developed by applying the method to a sample problem that is representative of the system to be developed.

**Management Issues:** Determining the support provided by the method to those who must manage the development process as well as the costs and benefits of adopting and using the method. One should recognize that methods are used within particular organizations that have established ways of conducting business.

As suggested by Wood et. al. [150] it may not be possible to reduce the evaluation to a completely objective process supported by a list of questions with quantifiable answers. Such a reduction may result in an oversimplification of a difficult task.

In the preliminary analysis described above we have attempted to provide some insight into the way an organization-oriented approach would be applied. We have attempted to show where the advantages or problems may appear. In particular we have addressed the issues of Need Analysis and Constraint Identification. We described the provisions in the specification language for describing complex distributed systems and the complexities introduced when using such specifications.

Wood et. al. [150] suggest a set of criteria that can be used to evaluate the Need Analysis and Constraint Identification. Examples of such criteria include: representation of input and output, representation of system behavior, representation of system structure, representation of system state and transitions, modularity, information hiding, abstraction, etc. A complete analysis of the organization-oriented approach would attempt to evaluate it according to all of the proposed criteria.

**User Requirements**

As described in Chapter 3 we were particularly interested in a language that will allow developers and researchers to precisely specify a real-time embedded distributed system and
to investigate different characteristics of such systems. We were particularly interested in a specification language that is expressive enough to cover real-time embedded distributed coordination and decision making techniques.

The main objective has been primarily addressed by providing a number of formal models of a real-time embedded distributed system. In particular we have extended known mathematical models to allow for additional features of such systems, namely, joint mental attitudes, social mental attitudes, and team structure.

As to the expressiveness of the specification language, this has been addressed in two ways. First, we have provided a hierarchy of models of real-time embedded systems. Each level in the hierarchy is a generalization of the previous level. Each level allows for the explicit specification of additional aspects of the distributed system.

Second, each of the models developed included a basic model and a set of axiom systems that characterize or constrain the behavior of the system. Each level in the hierarchy allowed us to explore different aspects of a real-time embedded distributed system.

The aspects of a real-time embedded distributed system include: (1) the components of the system; (2) the independent behavior of each component; and (3) their coordinated behavior. The details also include the division of tasks between the components, the capabilities, knowledge, and resources available to each component, the way options are evaluated and decisions made in a distributed setting, and many more. As part of this thesis we provided the Organization-Oriented specification language. This specification language allows the developer to explicitly specify all these aspects.

As suggested by Wood et al. [150] understanding the user requirements is best developed by applying the method to a sample problem. This has been achieved through the use of the Air-Mission Modelling problem. It is clear that applying the method to other problems would provide a further understanding.

Management Issues

Management issues are highly dependent on the particular developers and organization that would consider adopting the organization-oriented approach. Furthermore, given that the approach is at its early stages there is no empirical data or past experience to support such an evaluation.

Despite this lack of data and experience on the use of the organization-oriented approach one can partially address the following questions (suggested by Wood et al. [150]): Can one rapidly develop high-level design representations that can be analyzed to determine the most feasible design approach? Does the method help partition the system into manageable pieces that can be given out to individuals?

The organization-oriented approach provides a form of abstraction that allows the developers to develop the architectural design of a distributed system without the need to provide all the details of the components of the system. Furthermore, the interactions between these components can be explicitly represented. Given that the developers can determine the level of detail they provide this also helps partitioning the system into manageable components.
10.5. EVALUATION OF THE ORGANIZATION-ORIENTED APPROACH

There are many other aspects of managing software development that uses the organization-oriented approach. Such aspects include the time and cost of development, the level of training required, support for maintenance and extensions to the developed system, etc. Unfortunately they can not be answered without gaining further experience in using the method, measuring the development process, and analysing the collected data.
Chapter 11

Guided Team Selection

As part of the investigation of team-oriented systems and the operations of a team we have also investigated the problem of team selection. Team selection, the process of selecting a group of agents with complementary skills to achieve a goal, is an important collaborative task in multi-agent systems. Typically, team selection occurs at run-time using a first principles approach, for example after agents have exchanged relevant information about their abilities, loads, or other status. In time-critical domains such approaches may be impractical. Our work assumes that agents have limited resources and are embedded in a continuously changing world. In this chapter we provide a mechanism whereby system developers can describe “recipes” for team selection in terms of the required abilities of the team, and appropriate run-time constraints.

Note that this investigation is directed towards teams, that is, there are no organizational structures. Furthermore, we take the view here that teams are composed of sets of agents. Nevertheless, we believe that the approach presented here could be extended to organizations. In particular one could also consider the organizational structures known to the organizations when selecting the best organization to achieve a goal.

11.1 Introduction

The Artificial Intelligence and Multi-Agent Systems research communities have given much attention to the problems of planning from first principles for single and multi-agent systems. Given a particular initial state and a set of actions, planning from first principles involves generating a combination of such actions that, if executed, will attain a desired state. In the multi-agent setting, planning from first principles extends to the generation of a plan for multiple agents. Generating a multi-agent plan encompasses the combination of actions, the coordination required between them, and the selection of agents that will execute each of these actions.

In the past several years, the focus of research into single agent planning has shifted from using first principles for plan generation and plan recognition to the pre-planned approach to plan generation and plan recognition [49, 84]. A similar shift is occurring in
the focus of research into multi-agent planning [52, 69]. Two main assumptions distinguish the pre-planned approach from planning from first principles: (a) the environment in which the agents are situated is continuously changing; and (b) the agents have limited resources available to them for achieving goals. In the case of multi-agent planning there is an additional assumption: (c) the nature of the environment and the various agent's abilities are such that the agents are required to collaborate in order to achieve their goals. Such assumptions required the development of specialized techniques for guiding the agents in their individual and collaborative activities.

One technique involved viewing a plan as a “recipe” for achieving particular desired states [49]. Recipes are abstract combinations of actions and sub-goals and are provided to the agents in advance. They are used to guide and constrain the resource-bounded agents in their decision-making and coordination processes. Thus they reduce the time required for searching through a possible solution space and the communication required for performing cooperative activity.

Much of the attention of multi-agent pre-planned approach has been given to the problems of generating a combination of actions and coordination activity required for multi-agent behavior. The problem of selecting appropriate teams (from all the teams in the multi-agent system) to perform an activity has been mainly done using classical search techniques. Although some of these approaches attempt to reduce the complexity of the problem by either fixing the sets of agents that can be considered [27], providing some limitations on the possible agents that should be considered [129], or performing some of the selection at development-time [127]¹, none have attempted to use recipes to guide the selection process.

We refer to the use of recipes to guide and constrain the selection process as Guided Team Selection and to the recipes as allocations. These allocations are based on specifications of required abilities of the teams that could be selected. The specification takes the form of sets of goals or actions that are required. We refer to such a set as a role. Given a role one could identify the teams that have the ability to achieve the goals or execute the actions specified in the role. We would say that that team can fill that role.

Guided team selection is applicable to domains where there are multiple agents with limited resources that must collaborate to achieve their goals in an environment that is continuously changing. Guided team selection also makes a number of additional assumptions: (a) the construction of the team-oriented system allows for development-time processing; and (b) all teams and their abilities are known at development-time.

In this chapter, we focus on the use of allocations to guide the team selection process. In particular we present development-time algorithms that use allocations to generate a set of teams of agents for each goal that should be considered at run-time. We also provide a description of how the selection process proceeds during run-time. The formulation and analysis of these algorithms provide an insight into the benefits, limitations, and complexities of our approach.

¹We refer to the time during the construction of the multi-agent system as development-time and the time during execution as run-time.
The guided team selection approach allows the developer to determine the level of guidance. Such guidance can range from being fully guided, resulting in an approach similar to that taken by Decker and Lesser [27], to being fully unguided, resulting in an approach similar to that taken by Smith and Davis in the Contract-Net Protocol [115].

Although the pre-planned approach initially attracted some criticism, it has been successfully applied in its single and multi-agent forms to a number of complex problem domains. These problem domains include modelling of air-combat pilots [130], air-traffic management [74], and business process management. In such problem domains it was possible to describe the required behavior of agents as plans that were a direct formulation of the standard operating procedures as identified by domain experts. These plans were then used by the agents to react to the changing world and cooperatively achieve their individual and team goals within the boundaries of their limited resources.

It was identified that in such problem domains there is also a need for the agents to be able to form and dismantle teams in response to the changing environment and the introduction of new goals. Although such activity is done in a structured and well defined way, it has to be done under the constraints imposed by limited resources. These requirements have been the main motivation for the work described here. Furthermore, we believe that this work is applicable to similar classes of problems that have previously been addressed by reactive planning.

### 11.2 Roles and Allocations

We will be introducing agent and team roles as abstract specifications or “types” to enable the developer to provide useful information to limit the search space when the system is seeking actual agents/teams to fill various roles. Before defining roles and allocations, let us first define the notion of agents and teams adopted here. An actual agent is defined in terms of sets of actions, beliefs, goals, plans, and intentions [88]. The multi-agent system is a collection of actual agents. In this chapter we assume that the set of actual agents in the multi-agent system is known at development-time. An actual team is defined to be an ordered set of agents or actual teams. We refer to the elements of an actual team as actual sub-teams. An agent is thus regarded as the trivial case of an actual team.

One can view the process followed by an actual agent when determining how to achieve a goal as a means-end analysis with two distinct steps: (a) selecting a group of actual agents or teams that will attempt to achieve the goal; and (b) selecting the combination of actions to be taken by these actual agents in order to achieve the goal. The combination of actions is typically referred to as a plan. In this chapter the plans are multi-agent joint plans as described elsewhere [69]. Such plans include the knowledge required for the coordination of the group of actual agents when they attempt to achieve their joint goal.
11.2. ROLES AND ALLOCATIONS  CHAPTER 11. GUIDED TEAM SELECTION

11.2.1 Role Specification

A plan specifies the means by which a goal is achieved by an actual agent or team when certain preconditions are satisfied. Such means are specified as a labelled, directed, AND-OR graph. The labels specify: (a) a primitive action that is to be performed by an actual agent; or (b) a sub-goal to be achieved by an actual agent or team [69]. Each actual agent has a set of such plans referred to as the plan library of the agent.

An agent role is a specification of an abstract agent that possesses particular abilities or skills, i.e., goals and actions, that are characteristic of the defined role. For example, an electrician can be characterized by the ability to achieve specific goals, such as, restore power, disconnect power, etc. Hence the specification of an electrician can be viewed as the specification of an abstract agent. An actual agent in a multi-agent system can now be viewed as filling one or more agent roles.\(^2\)

We define a team role to be an unordered set of agent roles and other team roles with specific goals characteristic of the defined role of the team. For example, a manufacturing organisation can be viewed as a team role, with the CEO as an agent role and the marketing, production, and finance departments as other team roles. These team roles can be further decomposed into smaller team roles until we have only agent roles. The manufacturing organisation as a whole will have certain beliefs about its products, goals about its target production, plans about how to meet those targets, etc. We do not allow recursive definitions. A team can now be viewed as a specific instance of a team role.

A pure abstract team role includes a set of goals with a role variable instead of the whole set of sub-roles.

There are two main differences between an agent role and a team role: (a) team roles do not include actions, as we assume that all actions are taken at the individual agent level; and (b) team roles assign responsibilities to other sub-roles, thus introducing a notion of structure. Note that as with agent roles we can now define a hierarchy of team roles. If a group of actual agents in the multi-agent system, \(t\), is of a team role type, \(\rho\), then we say that \(t\) is an instance of \(\rho\) or that \(t\) can fill the role \(\rho\).

The specification of an actual agent or team as being able to fill a role depends on the skills of the actual agent or team. These in turn depend on the plans available to them. As the number of possible actual teams that can be formed is an exponential function of the number of actual agents in the system, it potentially can be a very large number.

It is not assumed that the developer of the system specifies the role filling abilities of all possible actual agents and teams. We simply allow the developer to specify the particular set of actual agents or teams that are able to fill each role. This information is then used during the automatic selection of an actual agent or team for a given goal. Note that if the developer does not provide any additional information, the system will automatically compute the set of teams that can fill a role.

\(^2\)One can also use normal set operations on agent roles to define other agent roles. Normal set operations are used by performing tuple-wise set operations. One can thus create a hierarchy of agent roles. At the top are roles that can achieve any goal and at the bottom specialized roles that can achieve a limited number of goals.

© Copyright Gil Tidhar, 1999 211
11.2. ROLES AND ALLOCATIONS  CHAPTER 11. GUIDED TEAM SELECTION

11.2.2 Allocation Specification

The specification of agent or team roles is static and can be reasoned about at development-time. Having this information is no guarantee of forming a team and executing the plans; there may be run-time constraints that prevent the formation of such a team. These run-time constraints are specified as part of the allocation. For each goal an allocation specifies, the form of agent or team roles required to achieve the goal and the beliefs concerning the state of the world or the team being considered that need to be determined at run-time. The developer can thus use allocations to specify run-time constraints on the state of the world and the state of the actual agents or teams that will be selected.

Given an allocation, we refer to the goal, the belief, and the role, as the relevance, team context, and potential team of the allocation respectively. Note that a particular goal may be the relevance of multiple allocations. Similarly, the same role may be the potential team of multiple allocations. Given a particular goal the developer can thus specify the roles that should be considered under different conditions.

Example 7

Assume that the developer has specified two roles, a Customer Service Representative (CSR) agent role and a Team Leader (TL) agent role (where TL and CSR are unique role constants). A Team Leader is also required to be a CSR, but with additional abilities, such as authorizing credit up to $200, instead of $50 permitted for CSR’s. Let \( \alpha \) be an actual agent of type CSR and one of its goals includes the goal of getting an authorization for crediting a customer’s account with $100. Since \( \alpha \) is a CSR it can not achieve this goal and it will have to select an actual agent or team that is able to do so.

In the absence of any role specifications, all actual agents and teams in the system are potential participants in the collaborative activity. However, if there exists an allocation which specifies that the role required for authorizing this amount is a TL, then the number of possible alternatives is reduced to agents of type TL. The developer can now create a goal for credit authorization and define two allocations, both relevant to this goal but with different team contexts and potential teams: (a) a team context where the amount is up to $50 and the potential team is a CSR; and (b) a team context where the amount is up to $200 and the potential team is a TL.

We can define the CSR agent roles to be: \( \text{CSR} = \langle \text{assure-service, report-fault, credit-customer}, \{\text{start-computer}\} \rangle \) and the TL agent role to be:

\[
\text{TL} = \langle \{\text{annual-review, produce-work-plan}\}, \emptyset \rangle \cup \text{CSR} = \langle \{\text{annual-review, produce-work-plan, assure-service, report-fault, credit-customer}\}, \{\text{start-computer}\} \rangle.
\]

Note that the TL is also a CSR but has some additional goals that it can achieve. To operate the exchange during one eight-hour shift (i.e., achieve the goal operate-shift) one actual agent of type TL and four actual agents of type CSR are required. The developer can thus define an allocation, CSA, that has as its potential team a team role including these requirements. Here we define this role explicitly and name it Customer Service Shift (CSS). The CSS role is defined as:

\[
\text{CSS} = \langle \{\text{operate-shift}\}, \{\text{TL}\}, \{\text{CSR}\}, \{\text{CSR}\}, \{\text{CSR}\}, \{\text{CSR}\} \rangle.
\]

The CSA allocation will thus be defined as:

\[
\text{CSA} = \langle \{\text{operate-shift}\}, \{\text{operate-shift}\}, \text{CSS}, \text{CSS} \rangle = \langle \{\text{operate-shift}\}, \text{CSS} \rangle = \langle \{\text{operate-shift}\}, \{\text{operate-shift}\}, \text{CSS}, \{\text{operate-shift}\}, \text{CSS} \rangle.
\]

Note that since a TL is also a CSR, then according to the definition of CSS, an actual agent that can fill the TL

© Copyright Gil Tidhar, 1999 212
11.3 Team Selection

Let us now describe how the specification of roles and allocations are used to guide the process of team selection. The selection process starts with a given goal to be achieved (see Figure 11.1, where highlighted arrows indicate guided team selection approach). For the given goal we identify the set of allocations that are relevant to it. Each such allocation describes a role and a set of run-time constraints on the selected team. Given a particular relevant allocation, we can identify the set of actual teams able to fill the specified role. Each such actual team is referred to as a relevant team for the given goal. Each relevant team that satisfies the run-time constraints of the relevant allocations is referred to as an applicable team for the given goal. If the set of applicable teams is empty then the attempt to achieve the goal fails; otherwise, one of the teams in the set is selected, formed, and is delegated the goal to be achieved.

The process of team formation has been addressed in previous work [127] and is beyond the scope of this work. Note however that if the selected team cannot be formed then another team from the set of applicable teams should be selected. If none of the teams in the set of applicable teams can be formed then the attempt to achieve the goal fails.
11.3. TEAM SELECTION

11.3.1 Relevant Teams

Given a particular goal \( g \) and a team \( t \), \( t \) is relevant to the goal \( g \) if and only if there is an allocation \( \alpha \) that is relevant for \( g \) and \( t \) is capable of filling the team role specified as the potential team of \( \alpha \).

If the developer does not specify the roles that the teams in the system can fill then the system will automatically compute the set of roles that each team can fill. Determining whether a team can fill a particular role will depend on the team’s ability to achieve the goals specified in the role. This in turn depends on the availability of allocations relevant for that goal and the ability of the actual sub-teams to fill the roles as specified in the potential team of each allocation. Given a goal \( \gamma \), the function \( \text{allocs}(\gamma) \) returns the set of allocations with a relevance that unifies with \( \gamma \). The set of teams that are relevant to an allocation \( \alpha \) is computed using the algorithm \text{alloc-relevant-teams} and is denoted \( AR_\alpha \).

In the following algorithm, the set \( S_\alpha \) is a cross-product of the sets of teams that can fill the roles specified in the potential team of the allocation (denoted as \( \text{potential}(\alpha) \)). It is computed using the algorithm \text{role-filling-teams}. Since the teams that are considered are required to achieve a goal we require that the team possesses at least one plan for achieving this goal. Hence, in the following algorithm the set \( AR_\alpha \) is the subset of teams in \( S_\alpha \) that also possess a plan for achieving the relevant goal of the allocation \( \alpha \) (denoted as \( \text{relevance}(\alpha) \)). The function \( \text{plans}(\tau) \) returns the set of plans that a team \( \tau \) possesses and the function \( \text{purpose}(p) \) returns the relevant goal of plan \( p \).

\[
\text{alloc-relevant-teams}(\alpha) \\
S_\alpha = \text{role-filling-team}(\text{potential}(\alpha)) \\
AR_\alpha = \{ \tau \in S_\alpha : \exists p \in \text{plans}(\tau) : \text{Unify}(\text{purpose}(p), \text{relevance}(\alpha)) \} \\
\text{return}(AR_\alpha)
\]

Given a team role \( \rho \), the algorithm \text{role-filling-teams} computes the set of teams that can fill the role \( \rho \) (denoted by \( RF_\rho \)). This algorithm is based on the set \( AR_\alpha \) that is computed using the algorithm \text{alloc-relevant-teams}. The function \( \text{actperf}(a) \) returns the set of actual agents (also defined as singleton teams) that can perform \( a \); the function \( \text{expreq}(\rho) \) returns the set of requirements for role \( \rho \) (i.e., the goals and/or actions); and the function \( \text{teams}(\rho) \) returns the set of developer defined actual agents or teams that can fill this role.

\[
\text{role-filling-teams}(\rho) \\
\text{if } \rho \text{ is an actual team then} \quad \text{return}(\rho) \\
\text{if } \rho \text{ is a pure abstract team role then} \\
\quad \sigma = \text{first}(\text{expreq}(\rho)) \\
\quad \text{if } \sigma \text{ is an action then } RF_\rho = \text{actperf}(\sigma) \\
\quad \text{if } \sigma \text{ is a goal then} \\
\quad \quad RF_\rho = \bigcup_{\alpha \in \text{allocs}(\sigma)} \text{alloc-relevant-teams}(\alpha)
\]
otherwise
   if $\text{teams}(\rho) \neq \emptyset$ then
      return($\text{teams}(\rho)$)
   $RF_{\rho} = \times_{v_i \in \text{subteams}(\rho)} \text{role-filling-teams}(v_i)$
for $\sigma \in \text{expreq}(\rho)$ do
   if $\sigma$ is an action then
      $RF_{\rho} = RF_{\rho} \cap \text{actperf}(\sigma)$
   if $\sigma$ is a goal then
      $RF_{\rho} = RF_{\rho} \cap \bigcup_{\alpha \in \text{allocs}(\sigma)} \text{alloc-relevant-teams}(\alpha)$
return($RF_{\rho}$)

Note our assumption that all the required knowledge, e.g., actual agents, available allocations, available plans, etc., is known at development-time. Also, it is assumed that all actual agents and all allocations are known to all other actual agents. This assumption can be relaxed by introducing the notions of an allocation library and a known agents library to each actual agent. These libraries will hold the list of known allocations and known actual agents respectively. We can then modify the algorithms to consider only allocations and actual agents from each actual agent’s libraries.\footnote{Such an approach will thus give a new meaning to the saying “It does not matter what you know but rather who you know.”}

The algorithms used for this computation are similar to the algorithms goal-achieving-teams and plan-achieving-teams used in previous work [69].

11.3.2 Applicable Teams

The team context of an allocation specifies the state of the world or the state of the potential team that should hold for the team to be considered applicable for the goal. This allows the developer to guide the system in the dynamic aspects of the selection of teams and further restrict the number of teams considered. More formally, we say that a team $\tau$ is applicable to achieve a particular goal if and only if it is relevant with respect to an allocation $\alpha$ and the team context of $\alpha$ is true. The choice of the team will now be made from the set of applicable teams.

The availability of the information regarding the state of the team depends on the way this information is disseminated between the actual agents. The dissemination of the state of a team to other teams ensures that when an actual agent is required to reason about that state, it has the required information available. This information is made available to the actual agent through its beliefs about the world. The way each predicate is maintained and where the information is stored depends on the implementation of that predicate. One approach may be to retrieve the information from a local knowledge base that holds the model of the world. Another option is to query other actual agents’ knowledge bases when the information is required. The method for retrieving the information should be defined for each component of the model of the world. A similar consideration should be given...
to changes in the state of the team, such as adopting a new goal. The developer should specify what the behavior of the system should be when the state changes. The behaviour will be determined by the state dissemination algorithm that is adopted.

The problem of information dissemination has been well studied by the distributed systems community [2, 83]. Distributed approaches include: information distributed on request; local information diffused between agents; and bidding techniques. Other approaches take a more “centralized” view, with the communication managed by a central process. Discussion of these issues is not pertinent to our focus here on team selection, beyond noting that performance will be affected by the actual choice.

Example 8
Consider the previous example where the developer imposes the restriction that in the allocation CSA the Team Leader cannot fill the role of a CSR. Now consider a particular multi-agent system that has been developed using these specifications. In this system there are two actual agents of type TL, t11 and t12, six actual agents of type CSR, csr1,...,csr6, and no specific information on the teams that can fill the role CSS. Also, all actual agents have plans for operating the exchange.

Given the goal operate-shift the only relevant allocation is CSA and the relevant teams are a combination of all teams that are composed of one of the TLs and four CSRs. The set of relevant teams, computed by role-filling-teams is a cross-product of all the teams that can fill each of the roles in the potential team of CSA.

On the other hand, if the developer had specified that CSS can only be filled by the teams teams(CSS) = \{t1, csr1, csr2, csr3, csr4\}, \{t1, csr1, csr2, csr3, csr5\}, \{t2, csr1,csr2, csr3, csr5\}, then there is would have been no need to compute this set and the team selection would have been trivial. ■

11.4 Analysis
Agent and team roles were introduced as “types” to enable the developer to reduce the part of the means-end tree that the system searches when seeking actual agents/teams. In this section we provide an analysis of the processes described above. Furthermore, we analyse the effects that the typing of actual agents and teams has on the breadth and depth of the means-end tree.

11.4.1 Formulation of Processes
For each goal, the number of relevant teams is a function of: the number of relevant allocations; the number of sub-roles in each allocation; and the number of teams that can fill each role required for each allocation. The teams that can fill each role are either provided by the developer or have to be computed using the role-filling-teams algorithm. This computation depends on the number of indirect recursive calls and the number of teams that can perform each action. The number of recursive calls depends on the information...
provided by the developer about the teams that can fill a particular role and the occurrence of actions in the definition of these roles. Whenever the developer provides more information, the average number of calls is reduced. When the developer provides all the information needed, there will be only one call to the algorithm.

We assume for simplicity that in each allocation each sub-role of the potential team is required to achieve only one goal. We then define the following domain-dependent parameters:

1. the average number of relevant allocations for each goal is \( l \);
2. the average number of sub-teams in the potential team of an allocation (each required to fill a role) is \( m \);
3. the average number of calls to the role-filling-teams algorithm is \( f \);
4. the average number of actual agents that can execute a particular action is \( n \); and
5. the average proportion between: (a) the number of actual teams that are relevant for an allocation and also possess the plan in which this goal appears; and (b) the number of actual teams that are relevant for an allocation, is given by \( 0 \leq q \leq 1 \).

Note that the parameter \( q \) is required because we assume that joint plans include knowledge about the coordination required for achieving a joint goal. Using such an approach reduces the communication required for coordination but it also constrains the teams that can achieve a goal to those that possess an appropriate joint plan. Using the role-filling-teams and alloc-relevant-teams algorithms the average number of relevant teams for a goal \( \sigma \), denoted \( R_\sigma \), is:

\[
\text{average}(R_\sigma) = l \times |\text{alloc-relevant-teams}(\alpha)|
\]

(11.1)

The algorithm alloc-relevant-teams calls the algorithm role-filling-teams on the potential team of \( \alpha \). We thus get that based on assumption (2) above if \( \sigma_i \) is a goal assigned to the sub-role \( v_i \) in the potential team of \( \alpha \) then the average number of relevant teams for the goal \( \sigma \) is:

\[
\text{average}(R_\sigma) = l \times (|\text{role-filling-teams}(v_i)|)^m
\]

(11.2)

Based on the assumption (5) above we get that the average number of relevant teams for a goal \( \sigma \) is:

\[
\text{average}(R_\sigma) = l \times (q \times |R_{\sigma_i}|)^m
\]

(11.3)

Since the average number \( |R_{\sigma_i}| \) is equal to \( \text{average}(R_\sigma) \) we get that the average number of relevant teams for a goal \( \sigma \) is: 
11.4. ANALYSIS

CHAPTER 11. GUIDED TEAM SELECTION

\[
\text{average}(R_\sigma) = l \ast q^m \ast (l \ast (q \ast |R_\sigma|)^m)^m = (11.4)
\]

\[
= l \ast q^m \ast l^m \ast q^m \ast (|R_\sigma|)^m^2
\]

Based on assumptions (3) and (4) above we get that the average number of relevant teams for a situation \( \sigma \) is:

\[
\text{average}(R_\sigma) = l \ast q^m \ast l^{-1}(m^i) \ast q^m \ast l_{(m^i \ast d_i)} \ast n^{(m^f)} (11.5)
\]

If we make equation (11.5) hold for the continuous case of \( f \) we get the following equation:

\[
\text{average}(R_\sigma) = l \ast f^{(m^i \ast d_i)} \ast q^m \ast l_{(m^i \ast d_i)} \ast n^{(m^f)} (11.6)
\]

Note that this number can decrease if sub-roles are required to achieve more than one goal in an allocation. Since the set of relevant teams is created in a constructive way, we need only add the complexity of: (a) determining the set of relevant allocations for a given goal; and (b) determining for each allocation the roles each actual sub-team is required to fill. Both (a) and (b) can be done once. If there are \( p \) allocations defined and the parsing of each allocation is \( k_1 \), then the complexity of (b) is \( k_1 \ast p \). If there are \( q \) goals referred to in the system and the complexity of the unification function \( Unify \) is \( k_2 \), then the complexity of (a) is \( k_2 \ast p \ast q \). If we also make equation (11.5) hold for the continuous case of \( f \), we get that the average total complexity of calculating the relevant teams for a given goal is:

\[
l \ast f^{l-1(m^i \ast d_i)} \ast q^m \ast f_{l^i \ast d_i} \ast n^{(m^f)} + p \ast (k_1 + k_2 \ast q) (11.7)
\]

The complexity of calculating the set of applicable teams depends on the complexity of the run-time constraints specified in the allocation and the complexity of the state dissemination algorithms. The analysis of these algorithms is beyond the scope of this work.

11.4.2 Effects on Means-End Tree

Given a particular goal, the part of the means-end tree that is related to the process of team selection has at its leaves the set of applicable teams. The computation involved in exploring this tree depends on the typing of actual agents and teams done by the developer. This dependency is reflected in the number of recursive calls to the role-filling-teams algorithm (i.e., \( f \)). It is also dependent on the specialization of the actual agents and teams reflected in the number of teams that can execute a particular action (i.e., \( n \)).

Let us now describe a few examples that will demonstrate the effects of reducing the depth and breadth of the means-end tree on the number of relevant teams. If, for example, the roles are defined only in terms of actions, then the algorithm is called only once and the depth of the tree is 1, i.e., \( f = 1 \). On the other hand, if for each role the developer provides the set of teams that can fill this role, then again the algorithm is called only once. In this case, we get that the number of relevant teams is:
Analysis

\[ \text{average}(R_\sigma) = l \cdot (q)^m \cdot (n)^m \] (11.8)

Figure 11.2: Average relevant teams per goal (with \( l = 2, q = 1, m = 2 \)).

If we assume that the average number of relevant allocations is 2 (i.e., \( l = 2 \)), that each relevant team has a matching plan (i.e., \( q = 1 \)), that the average number of teams that can execute a particular action is 2 (i.e., \( n = 2 \)), and that the average number of sub-teams in each allocation is 2 (i.e., \( m = 2 \)), then we get that the average number of relevant teams per goal is only a function of the amount of typing done by the developer and the average number of agents that can fill each role. The formulation of this is (see Figure 11.2 – dashed line – for \( f \in [1-2.5] \)):

\[ \text{average}(R_\sigma) = 2^{(f_0^{-1} 2^i \cdot di)} \cdot 2^{(2f)} \] (11.9)

Let us now examine the effect of changing the specialization of the teams when the number of recursive calls is fixed. If we assume that the number of such calls is 2 (i.e., \( f = 2 \)), that the average number of relevant allocations is 2 (i.e., \( l = 2 \)), that each relevant team has a matching plan (i.e., \( q = 1 \)), and that the average number of sub-teams in each allocation is 2 (i.e., \( m = 2 \)), then we get that the average number of relevant teams per goal is only a function of the number of teams that can execute a particular action. The formulation of this is (see Figure 11.2 – filled line for \( -n \in [1-2.5] \)):

\[ \text{average}(R_\sigma) = 2^{(f_1 2^i \cdot di)} \cdot n^{(22)} \] (11.10)

One can now use the above formulation (Equation 11.5) for the continuous case and compare the effect of the various parameters on the number of relevant teams. It can also
be used to provide estimates of the complexity of the process and the expected response
time of the system. For example, let us compare two possible scenarios: (1) the number of
recursive calls is fixed to 2 (i.e., \( f = 2 \)), the number of teams that can execute a particular
action is in the range \([1–2.5]\) (i.e., \( n \in [1–2.5] \)) and all other parameters are fixed; and (2)
the number of teams that can execute a particular action is fixed to 2 (i.e., \( n = 2 \)), the
number of recursive calls is in the range \([1–2.5]\) (i.e., \( f \in [1–2.5] \)) and all other parameters
are fixed with the same values as before.

When analyzing these two scenarios, we get that the number of recursive calls is domi-
nant in the process and should be minimized as much as possible. That is, although difficult
to do, the developer should focus efforts on attempting to provide as much information as
possible on the actual agents and teams that can fill each role.

### 11.5 Comparison with Relevant Work

Team selection can be done by actual agents exchanging, at run-time, full information
about their skills, goals, plans, or beliefs. Probably the best known selection method
in Multi-Agent Systems is the Contract-Net Protocol [115] (CNP). Given a goal to be
achieved, it is first decomposed into sub-goals and the protocol then uses a bidding-like
mechanism to select the actual agents that will attempt to achieve the given sub-goals.
Such a selection process either involves an exponential number of possible actual agent
combinations, where all processing is done at run-time.

Note that in the CNP if a sub-goal has to be further decomposed, then the selection
process will have to be repeated for that sub-goal. Therefore we can view the parameters
from the formulation in the previous section, \( l, m, f, \) and \( n, \) as meaning: the number of
decompositions of each goal; the number of sub-goals each goal is decomposed into; the
number of times a goal has to be decomposed before it is decomposed into primitive actions;
and the average number of agents that can execute a particular action respectively.

Thus, there are at least four major differences between the two approaches: (1) the
actual values of the parameters; (2) the time in which the relevant teams are determined;
(3) the additional overhead required; and (4) the flexibility of the approach.

With respect to the values of the parameters, we argue that in the Guided Team
Selection (GTS) approach, such values would be less than the values used in CNP. The
reason being that in GTS, the developer has to specify the relevant allocations for each
goal. In CNP, each goal is decomposed into sub-goals. When specifying the allocations,
the developer restricts the possible means that can be employed to achieve the goal to the
most likely ones. It seems then that in general this number will be substantially less than
the number of possible decompositions of a goal into sub-goals.

With respect to the time in which the relevant teams are determined, GTS does this
at development-time while in CNP, this is done at run-time. This gives GTS an obvious

---

\(^4\)The parameter \( q \) is not relevant for CNP, since coordination is centralized (i.e., done by the announcer) and thus achievement of goals depends on communication not on common knowledge of a joint plan. We can thus say that \( q = 1 \).
advantage when considering time-critical domains. Note however that like in CNP, the GTS approach makes the final selection run-time. This allows the selecting agents to take into account dynamic consideration such as workload or the cost of achieving the goal.

With respect to the additional overhead required, CNP determines the set of relevant teams is done through the use of communication. The use of communication is typically very expensive, both in time and actual cost of usage. With respect to flexibility, CNP is more flexible in some respects than GTS. In CNP there is no substantial cost in adding or removing actual agents unexpectedly during run-time. This flexibility does not exist in GTS, since it is assumed that the actual agents are known at development-time.

Unlike CNP, in the approach adopted by Decker and Lesser [27] agents are homogeneous and teams are fixed in advance. Given a particular goal/task, team selection is reduced to the problem of load-balancing, i.e., dynamically assigning the goal/task to one of a small number of known teams. Obviously, this approach avoids the complexity of team selection. Unfortunately it also imposes a limitation on the ability of the actual agents to respond to the changing world by dynamically forming new teams or re-grouping.

Another variation of the CNP is the work of Shehory and Kraus [102]. Influenced by Game Theory, the solution they provide is a combination of the CNP and a Game Theoretic approach. In their work, they assume that: actual agents are capable of achieving the goals without assistance from other actual agents; cooperation emerges as a result of the actual agents attempting to maximize some individual utility function (i.e., it costs less to cooperate); and there is no central design process.

In previous work [127] we presented a solution to the team selection problem that is performed at development-time. Like the CNP, it is fully unguided and resulted in an exponential number of teams. An analysis of these solutions can be found elsewhere [124].

An unguided team selection results in either a blow-out in the number of interactions required to select the members of a team or a blow-out in the number of teams considered. Obviously this is unacceptable in time-critical domains where the actual agents have limited resources. As mentioned above, the GTS approach provides a generic way for the developer to provide additional knowledge to the system, so that this number can be substantially reduced. It also allows the developer to determine the level of guidance (or restriction) that is used in the team selection process.

11.6 Conclusions

The process of selecting the means for achieving a goal in a multi-agent system can be decomposed into two main steps: the first is the selection of a team that will attempt to achieve the goal; and the second is the selection of a (joint) plan of action. The choice of teams that can potentially achieve a goal depends on the skills of each team. In a totally unrestricted system, the number of such teams may be very large and hence such an approach is unsuitable for time-critical domains.

In this chapter we have adopted an approach similar to that taken by reactive planning. We have defined a mechanism for team selection that allows the developer of the system
11.6. CONCLUSIONS

CHAPTER 11. GUIDED TEAM SELECTION

to guide the process by specifying team selection “recipes”. These recipes take the form of allocations. For a given goal, an allocation specifies the types of actual agents and teams that could be selected and appropriate run-time constraints.

These specifications are then used by algorithms that generate the appropriate sets of relevant teams at development-time. This computation reduces the run-time complexity of team selection and is therefore more suitable for time critical domains.

After selecting a team, one must form the team and delegate the goal to it. This raises issues such as the synchronization of the mind-set of team members [127]. Similarly after selecting a plan, the team has to execute it. Executing joint plans is an area that has received much attention. Such work includes the work of Durfee and Lesser [34], Grosz and Kraus [52], Kinny et al. [69] and many others.

The theoretical work described in this chapter has been implemented as two separate demonstrator systems using different technologies [44]. The first demonstrator has used the dMARS™ agent-oriented system [29] as a basis for adding a team selection process. The second demonstrator has used the functional language, Gofer [66], for implementing the agent architecture and the team selection process. These implementations have confirmed that the major benefit from the approach described here is obtained when the developer specifies the actual agents and teams that can fill some of the roles and when the actual agents are fairly specialized, i.e., there is not much overlap between their expertise.

Goals adopted by an actual agent or team would typically lead to the adoption of sub-goals. Further optimization of the selection process can be achieved by allowing the developer to specify, where possible, the types of teams that should be considered for achieving the sub-goals. This information, together with the allocations, can then be used in the selection process. We leave this as future work.

If we allow the developer to specify the types of teams that should be considered for sub-goals and if we enforce that sub-goals can only be achieved by sub-teams of the team achieving the goal, then we can provide some mechanisms for checking the plans that can be used by different teams. Again this is left as future work.

As mentioned above the approach described here could be extended to take the selection of organizations. One could extend the algorithms to take into account the organizational structures known to organizations. This information will be used to selecting the best organizations that could achieve a given goal.
Chapter 12

System Design

This chapter describes the design of a prototype system for developing Organization-Oriented system. It is in essence a development environment. We refer to this development environment as the TOP system. We also provide a description of an analysis and design approach which results in a specification of an organization-oriented system that could be built using this development environment.

The design of the TOP system is based on the concepts and models described in Chapter 8 and Chapter 10. In particular the design focuses on the concepts of Command, Control, and Communication.

The system itself is an extension of the dMARS™ system developed by the Australian Artificial Intelligence Institute [29]. dMARS™ is an Agent-Oriented system and is an implementation of the BDI model developed by Rao and Georgeff [88]. A detailed description of the dMARS™ approach and architecture can be found elsewhere [29, 67, 85]. Before we proceed let us describe the administrative and technical constraints and assumptions made in developing the TOP system.

12.1 Constraints and Assumptions

Here we describe the assumptions and constraints under-which the design of this development and design chapter are to be considered. These assumptions and constraints can be generally divided into two categories: (1) administrative; and (2) technical.

Administrative Constraints and Assumptions

1. The system described here is only a demonstration system designed to explore the complexities and issues of command, control, and communication within a group of artificial agents.

2. The amount of resources (i.e., human resources and time) are extremely limited.

3. The code developed under this project should be distinctly identified and separable from any other software development.
4. The application domain in which this work is to be tested is the domain of air mission modelling [130].

Technical Constraints and Assumptions

1. The approach adopted here is aimed at the development of multi-agent systems that have to present predictable, reliable, and verifiable behavior.

2. The development of the demonstrator is to be done as an extension to the development of the dMARS™ system [29].

3. The demonstrator system is to be used in developing real-time distributed systems.

4. The user of the demonstrator system (i.e., the developer of an organization-oriented system) can provide the system with specifications as to the required command, control, and communication behavior.

12.2 High Level Design

The dMARS™ system implements a Belief-Desire-Intention (BDI) model of an agent. In this model an agent is constructed of a set of:

- **beliefs** about the current state of the world;
- **goals** that are future states of the world to be achieved;
- **plans** that are abstract combinations of goals to be achieved and actions to be performed and that describe how to achieve such goals; and
- **intentions** that are committed instances of such plans.

This model of an agent is provided with social knowledge and ability that allows it to be part of an organization of such agents. These knowledge and abilities are referred to as a **social layer**. The extended model of the agent is referred to as a **social agent**.

In the BDI approach one distinguishes between a plan and an intention, which is a committed instance of a plan. In a similar fashion we distinguish between an organizational structure and a committed instance of such a structure. The latter is referred to as an **organizational commitment**. The social layer includes information on:

**Organizational Structures**: includes knowledge about the operations within a particular organization and in particular the issues of command, control, and communications. This knowledge includes:

- the knowledge about who makes decisions within an organization and how such decisions are made, e.g., joint decision making protocol like voting;
• the knowledge on the flow of control of execution and the coordination of the joint activity, e.g., the existence of an organization coordinator; and

• knowledge about the exchange of information within an organization, e.g., the available communication links between the various agents and organizations and the communication protocols.

Social Awareness: includes knowledge about other agents and organizations. This knowledge includes:

• currently known agents and organizations, including organizations of which the agent is a member;

• organizational commitments, i.e., adopted organizational structures;

• the skills of known agents, including plan libraries. This also includes organizations of which the agent is a member; and

• the current state of known agents and organizations, including the mutual beliefs, joint goals, and joint intentions. This also includes organizations of which the agent is a member and the known social commitments between the various agents and organizations (including the agent itself).

Organization Formation: includes knowledge about the formation of new organizations. This knowledge includes:

• predefined allocations as to the types of organizations that should be considered for given goals. Known organizations that are of that type are referred to as potential organizations;

• algorithms for negotiating with potential organizations; and

• knowledge on performing mind-set synchronization.

Previous work has focused on the organization formation aspects [44, 128]. In the following sections we will show how that work fits within this framework and system design. Work on the social awareness has been only partially addressed in previous work [15, 44, 69, 128]. The main focus of this work is on the representation and implementation of the organizational structures and the related aspects of social awareness and how this information is used to affect the behaviour of the agent.

All the above describes the additional knowledge that is required for the development of a social agent. In addition to this additional knowledge we also consider the required changes to the plan language, the changes to the operational semantics of these plans, and the changes to the execution mechanism of the agent.

12.2.1 Organizational Structures

The knowledge about organizational structures takes the form of a library of the various abstract organizational structures [123, 125, 130]. The knowledge about each organizational
structure can be divided into knowledge about: (1) an organizational composition, i.e., the roles that are part of the organization; (2) organizational relationships, i.e., the command, control, and communication relationships that exist between the organization and the specified roles and between the roles themselves; and (3) goal achieving responsibilities, i.e., the allocation of responsibilities of achieving sub-goals to the specified roles.

Here we will mainly focus on the specification of the organizational composition and the organizational relationships. The organizational composition is basically a specification of the various role names that are part of the organizational structure, e.g., leader, wingman, controller, etc.

The organizational relationships are categorized in terms of the command relationships, control relationships, and communication relationships. The relationships correspond respectively to the following activities and processes that occur within an organization: (1) decision making processes (i.e., command); (2) coordinated activity (i.e., control); and (3) flow of information (i.e., communication).

We consider the Command aspects of organization behavior to be the way an organization delegates the achievement of goals to other organizations. In particular, it can be the decision made by one sub-team as the goals that will be achieved by the whole organization or the assignment of sub-goals to sub-teams. Thus the decisions made by an organization in command can be regarded as addressing the question of what goals should another organization achieve.

The issue of Control is directly related to the level of autonomy that the sub-teams have when achieving their assigned sub-goals. We consider the Control aspects of organization behavior to be the way an organization makes decisions with respect to the internal operations of other organizations when attempting to achieve assigned goals. These decisions include: (1) which goals should be achieved by which sub-teams (i.e., role assignment); (2) how the organization should achieve such goals, i.e., selecting amongst multiple joint plans; and (3) when to achieve each goal, i.e., scheduling multiple intentions towards achieving the organization’s assigned goals.

The issue of Communication within an organization is directly related to the issue of Control and how tightly are organizations coupled to each other. We consider the Communication aspects of organization behavior to be the way the organization exchanges information about the state of the sub-teams and the decisions being made. Thus the Communication aspects include: (1) the exchange of decisions; (2) the reporting of new or changing beliefs; and (3) the reporting of attempts to achieve the assigned sub-goals.

For each of these processes the organizations include a specification of the required sets of responsibilities (or roles). The behavior of the team members may be affected by the assignments of these responsibilities to the various sub-teams.

As an example, consider the domain of air mission modeling [130]. In this domain consider an eight-ship (i.e., a group of eight aircraft) that is composed of two four-ships each of which is composed of two pairs of aircraft. Each of the aircraft in a pair has an assigned set of responsibilities with respect to making decisions for the whole group, acting in a coordinated manner, and communicating decisions and information. Similarly, there is an assignment of responsibilities to the pairs that compose each of the four-ships and
the four-ships that compose the eight-ship.

Organizations are adopted (or are committed to) for particular goals. The choice of the organization depends on the particular circumstances under which the goal is to be achieved. Committed organizations can thus be viewed as part of the means towards achieving the goal. At any given time a organization may be committed towards achieving multiple goals. Hence at any given time it may also adopt (or commit to) a number of organizations that correspond to these committed goals. For example, a organization of aircraft may have the goal to perform a mission. In order to achieve this goal it may adopt an organization that may determine which aircraft is responsible for making tactical decisions, who should information be sent to, etc. As part of achieving this goal the organization may adopt the goal to fly in some spatial formation. To achieve the coordination required when flying in this formation the organization may also adopt a particular organizational structure that specifies which aircraft leads the organization, who coordinates the flight with whom, etc.

**Decision Making:** The knowledge about the decision making processes includes the responsibility of making the decisions for the organization. This responsibility includes the selection of joint goals that will be achieved. Furthermore, the knowledge about the decision making processes include knowledge about mechanisms or protocols for making joint decisions, knowledge on task decomposition, means-end analysis, task allocation, etc.

**Control of Coordinated Activity:** The knowledge about the flow of control when the agents perform a coordinated joint activity. This information includes the specification of information as to the mechanisms used for control, e.g., centralized coordinator, distributed coordination through common knowledge or continues exchange of information, etc. The particular protocols and mechanisms are specified.

**Flow of Information:** The knowledge about the mechanisms and protocols used to convey information between the team members are specified. It includes the allocation of responsibility to communicate and the sub-team to which information should be communicated. Furthermore, this knowledge includes the specification of mechanisms for the exchange of beliefs and mechanisms for the exchange of decisions.\(^1\)

### 12.2.2 Social Awareness

The social awareness of the agent includes all the information known to the agent about the agents and organizations that are part of the environment in which it is embedded. This knowledge can thus be viewed as a part of the agent’s beliefs about the world. Nevertheless we distinguish such beliefs as they are related to elements of the environment which are

---

\(^1\)Note that the exchange of decisions is part of the Flow of Information and not part of the Decision Making activity. This is because we separate the process of reaching a decision from the process of conveying it to other team members.
modelled as 'intentional’ entities, i.e., their behaviour is governed by some process of desire and intention \[9\]. The social awareness of the agent includes information about: (1) known agents and organizations; (2) committed structures; (3) skills and capabilities; and (4) current state.

**Known Agents and Organizations:** The agents or groups of agents that are part of an organization and that are known as such to the agent.\(^2\)

**Committed Organizations:** The organizational structures that are known to have been adopted and the corresponding allocation of roles to the various sub-teams are described. The knowledge about the committed structures of organizations can vary from complete knowledge to no knowledge at all. No knowledge of the organizational structure simply suggests that the agent only knows “about” the existence of the agent or organization.

**Skills and Capabilities** The skills of an agent or organizations are defined to be the goals it can potentially achieve and plans it can potentially instantiate and execute. The capabilities of an agent or organization are defined to be the goals it can actually achieve and plans it can actually instantiate and execute under the current circumstances. This component describes the known skills and capabilities of each known agents or organization. The knowledge about the skills and capabilities of agents and organizations can vary from complete knowledge to no knowledge at all.

**Current State:** The current state of an agent or organization is defined in terms of its current mutual beliefs, joint goals, joint intentions, and social commitments to other organizations. This component describes the knowledge about the state of each of the known agents or organizations. Such knowledge can vary from complete knowledge to no knowledge at all.\(^3\)

In the air mission domain the agents and organizations that are known to each of pilots in the eight-ship may include all the pairs, four-ships, and the eight-ship itself. The pilot assigned the role of the leader may also know about the existence of other aircrafts or organizations. The skills and capabilities may include the ability of the aircraft to attack ground targets or engage in air combat. A tanker does not have the skill to engage in air combat. Although a fighter aircraft may have the skill to engage in air combat it may have exhausted its weapons and thus does not have the capability of doing so.

\(^2\)Note that we thus enforce that an organization is simply a set of agents or sub-teams.

\(^3\)The knowledge the agent has about other sub-teams of organizations of which it is a member is affected by the way information is transferred between the sub-teams as defined in the specification of the flow of information of the currently committed organizations.
12.2.3 Organization Formation

The approach taken in this work is that organizations are formed to achieve particular goals or react to particular situations. It is thus the case that like plans, organizations are viewed as part of the means required to achieve particular ends. The knowledge available to the agent on the formation of organizations can be generally divided into three types based on the purpose of this knowledge. These types include: (1) allocations; (2) selection process; and (3) mind-set synchronization.

Allocations: This knowledge is related to the selection of an organization to be formed. The selection of an organization includes the identification of the organizations that have the potential to achieve a given goal. This information may obviously depend on the skills, capabilities, current state, and committed structures of the known agents and organizations. We refer to this type of knowledge as an allocation. Detailed discussion on organization selection can be found in Chapter 11. It is intended that the selection concepts, which have been implemented in a concept demonstrator [44], be integrated at a later stage into the design described here.

Selection Process: This type of knowledge is related to the actual interaction (e.g., negotiation) with the potential organizations and the final selection of the organization to be formed. In previous work we have suggested a number of algorithms that can be used in making this selection [127].

Mind-Set Synchronization: This type of knowledge is related to the establishment of the appropriate joint mental attitudes, i.e., mutual beliefs, joint goals, joint intentions, and committed organizations. The way in which such attitudes are formed is dependent on the organizational structure that should be adopted.

12.2.4 Plan Language

The knowledge stored in the various types of knowledge bases described above should be accessible from the agents procedural knowledge, i.e., the plans. Furthermore, changes to this information should affect the processing of the plans. Such changes should also trigger the creation of intentions to respond to these changes. For example, if a team member is removed the agent should respond to this change by initiating a change to the committed structure.

Some changes to the plan language have been suggested previously. Such changes primarily focused on: (1) providing coordinated execution of joint plans (i.e., Control) [69, 123]; and (2) supporting the organization selection process [44]. Here we focus on the changes to the plan language that are associated with supporting the Decision Making Processes (Command), Information Transfer (i.e., Communication), and the storage and

---

4If the organization has already been formed then the goal can be delegated to it.
access of this knowledge in the Organizational Structures. Changes to the plan language include: (1) changes to the invocation conditions; (2) access methods to the organizational structures; and (3) changes to the labels on arcs in the plan body.

Invocation Conditions
Changes to the invocation condition includes the addition of event types and changes to the structure of the goal events. New event types are introduced to allow for the identification of situations in which the organizational structure (and in particular the command, control, and communication relations) or social awareness are to be considered.

These situations include situations in which sub-teams that have been assigned roles that are in command have to make decisions on behalf of the whole organization. In this case the other sub-teams (that are not in command) need to be aware of such a situation and await the decision. Situations in which sub-teams have been assigned roles that are in control of the whole organization or of other sub-teams and have to make decisions for those sub-teams. The controlled sub-teams have to wait until the appropriate control decisions are conveyed to them.

The events should also be generated to indicate situations in which and agent becomes aware of the existence of a new agent or organization. A situation in which the state of a known agent or organization changes or that the organization has committed to another structure.

The structure of the goal events is changed with the addition of the organization this event is relevant for and the associated committed organization for which the event has been generated. This is particularly relevant with respect to new goals.

Organizational Variables
Organizational variables reference the roles in the committed structure and are denoted by the symbol “&”. For each role name defined in the structure we define a variable of that name preceded by the symbol “&”. This variable is bound to the organization that has been assigned that role. Thus if the committed organization that appears in the invocation condition has a role leader then one could use in the plans the variable &leader. Roles of other committed organizations can be bound in the normal way, i.e., by inspecting the organizational structure and binding variables accordingly (e.g., through unification).

Arc Labels
Changes to the structure of labels on arcs include the addition of: (1) the organization that is responsible for the execution of the label; and (2) the organizational structure that the organization should commit to. Both are optional. If the organization is omitted then the organization in the invocation condition is used. If the organizational structure is omitted then, if the organization is the same as the one in the invocation condition then the committed organization from the invocation condition is also used. Otherwise one of
the current relevant committed organizations for that organization is used. If there is more
than one committed organization then the structure-option event is generated.

12.2.5 Operational Semantics of Plans

Let us now describe the required changes to the operational semantics of the execution of
a plan. As with plans in the current dMARS™ system, here each plan is relevant to a
particular event. Such events are generated during the execution of plans (e.g., new-goal
event), changing the database (e.g., add-fact event), receiving external communication
(e.g., told event), or internal events associated the state of the execution and decision
making options (e.g., option events).

The changes to the operational semantics of the plan are directly associated with the
changes to the components of a plan. That is the changes to the invocation condition, arc
labels, and introduction of organizational variables.

A plan can only be executed by an organization with a committed organizational struc-
ture. The organization may be a singleton in which case all the roles in the committed
organizational structure are assigned to that agent. The posting of an event where either
the organization or the organizational structures are unbound results in the generation of
special events for selecting a organization and committing to an organizational structure.
This will be discussed in further detail in Section 12.2.6. Here we assume that the event
already specifies the organization and its committed structure and that an applicable plan
has been selected for this event.

As with dMARS™ only single agents actually process and execute plans. A successful
execution of a plan is a successful execution of all the arcs along a path from the distin-
guished start node to one of the end nodes. A successful execution of an arc is a successful
execution of the label on that arc. Note that the labels are now annotated with the or-
ganization that is responsible for the execution and the structure that that organization
should adopt.

The only change to the operational semantics of executing an arc from the semantics of
the dMARS™ system is with respect to the effect of annotating a label with a organization.
A label can be successfully executed only if the agent that is currently processing the label
is a member of the organization responsible for the execution. If the processing agent
attempts to process a label with a organization which it is not a member of then this
attempt fails immediately.

12.2.6 The Agent’s Control Mechanism

The changes to the agent’s control mechanism are mainly associated with the need to in-
corporate organization activity. In particular this is true with respect to the performance
of the decision making activity (i.e., command), coordination of joint activity (i.e., con-
trol), and exchange of information (i.e., communication). We say that a sub-team a is in
command of a organization A if and only if a has the command relation with the A and if
a is in command of $a$. Similarly we say that a sub-team $a$ is in control of sub-team $b$ if and only if $a$ has the control relation with $b$ and if $a$ is in command of $a$.

To support this activity the notion of committed organizational structure has been introduced. Thus the agent is now required to ensure that each event is responded to by the appropriate organization and that the organization has adopted the appropriate structure.

The basic changes to the agent’s control mechanism are with respect to the maintenance of multiple intention lists. Each such intention list is associated with the intentions of the organizations that the agent is a member of. Note that each agent maintains an intention structure for each organization it is a member of. The selection of competing intentions within a organization’s intention structure is managed by the organization and according to the organization’s organizational structure (i.e., the decision making responsibility). The selection of competing intentions across multiple organizations is managed by the agent itself. The potential conflicts between the agent’s commitments to different organizations can now be managed explicitly through the interactions between the different intentions.

Execution of an intention commences with the posting of an event. It continues in the following processing loop:

0) Select an event from the event list
1) **IF** event not tagged with organization **THEN**
   1.1) Post `sel-organization` event
   1.2) Goto (0)
2) **IF** executing organization not selected but in command **THEN**
   2.1) Post `delegate-command` event
   2.2) Goto (0)
3) **IF** organization not adopted structure **THEN**
   3.1) Post `sel-structure` event
   3.2) Goto (0)
4) **IF** event not tagged with plan **THEN**
   4.1) Get applicable plans
   4.2) **IF** no applicable plans **THEN**
       4.2.1) Mark goal as failed
       4.2.2) Post `goal-failure` event
       4.2.3) Post `goal-failure-communication` event
       4.2.4) Goto (0)
   4.3) **IF** more than one applicable plan **THEN**
       4.3.1) Post `organization-plan-option` event
       4.3.2) Goto (0)
   4.4) Select applicable plan for event
5) Form intention from event and plan
6) Add new intention to intention list of relevant organization

---

5This recursive definition will bottom-up in case the sub-team is an agent or if the command relationship is between the whole sub-team and itself.
7) IF no intention selected for organization THEN
   7.1) Get list of intentions for organization
   7.2) IF multiple intentions for organization THEN
       7.2.1) Post organization-root-option event
       7.2.2) Goto (0)
   7.3) ELSE
       7.3.1) Select intention for organization
8) IF no intention selected for executing organization THEN
   8.1) Get list of intentions for executing organization
   8.2) IF multiple intentions THEN
       8.2.1) Post root-option event
       8.2.2) Goto (0)
   8.3) ELSE
       8.3.1) Select intention for executing organization
9) IF no arcs selected for intention THEN
   9.1) Get next executable arcs
   9.2) IF no arcs (end node) THEN
       9.2.1) Mark intention as succeeded
       9.2.2) Post plan-success event
       9.2.3) Post goal-success event
       9.2.4) Post plan-success-communication event
       9.2.5) Post goal-success-communication event
       9.2.6) Goto (0)
   9.3) IF no arcs (all failed) THEN
       9.3.1) Mark plan as failed
       9.3.2) Post plan-failure event
       9.3.3) Post plan-failure-communication event
       9.3.4) Goto (0)
   9.4) Select next executable arc
10) IF arc labelled with action THEN
    10.1) Execute action
    10.2) IF action succeeded THEN
        10.2.1) Mark arc as succeeded
        10.2.2) Post action-success event
        10.2.3) Goto (0)
    10.3) ELSE
        10.3.1) Mark arc as failed
        10.3.2) Post action-failure event
        10.3.3) Goto (0)
11) ELSE
    18.1) Post event labelling arc
12) Goto (0)
12.2.7 Organization-Oriented Development Environment

The development environment is an extension to the development environment of the dMARS™ system. It includes the development of a new editor (named edit_soc) that allows the specification of the agent’s social aspects. This editor is similar in format to the dMARS™ form editors. The development environment will also include extensions to the dMARS™ process description and agent description editors (i.e., edit_pd and edit_ad).

The dMARS™ plan editor (i.e., edit_mpl) is also modified to accommodate the changes to the plan language described above. Similarly the plan compiler (i.e., dcomp) and control interface (i.e., dci) are modified as required.

12.3 Detailed Design

The detailed design of the organization-oriented system includes three main components: (1) the design of data structures that will be used to specify the organizational structures, social knowledge, and organization plans; (2) the design of an extension to the current dMARS™ interpreter to include social knowledge and the modified plan language; and (3) the design of a translation program that will transform the specification of the social agent (including revised plan language) into data structures interpreted by the interpreter.

The development environment will primarily be extended to add these structures. It is planned that the dMARS™ dgen facility will be used and that the editors will be automatically generated. Changes to the plan editor will be coded separately.

Organization Plan

The changes to the plan language to accommodate for organization activity include changes to the invocation condition and changes to the labels on the arcs. To minimize the changes to the current dMARS™ plan language we will use special symbols to indicate that a label on an arc is a organization-oriented label. Such an arc will be denoted by the special symbol “##”. The syntax in BNF of a organization arc label will thus be:

\[
<\text{GoalFormula}> ::= <\text{GeneralAchieveFormula}> \\
| <\text{GeneralTestFormula}> \\
| <\text{GeneralWaitFormula}>
\]

\[
<\text{OrganizationSymbol}> ::= ##
\]

\[
<\text{GeneralAchieveFormula}> ::= <\text{AchieveFormula}> \\
| <\text{OrganizationAchieveFormula}>
\]

---

This syntax should be read as a modification to with the dMARS™ plan language syntax.
<OrganizationAchieveFormula> ::= ( ! ( <OrganizationSymbol> 
   <OrganizationName> 
   <OrgStructName> 
   <SituationFormula> 
   ) 
   )

<GeneralTestFormula> ::= <TestFormula>
   | <OrganizationTestFormula>

<OrganizationTestFormula> ::= ( ? ( <OrganizationSymbol> 
   <OrganizationName> 
   <OrgStructName> 
   <SituationFormula> 
   ) 
   )

<GeneralWaitFormula> ::= <WaitFormula>
   | <OrganizationWaitFormula>

<OrganizationWaitFormula> ::= ( ^ ( <OrganizationSymbol> 
   <OrganizationName> 
   <OrgStructName> 
   <SituationFormula> 
   [<StopCondition>] 
   ) 
   )

<OrganizationName> ::= <CategOrVar>

<OrgStructName> ::= <CategOrVar>

The changes to the invocation condition includes the introduction of new events. To minimize the changes to the dMARS™ system these new events will be implemented as specialized add-fact events. The new events that will be generated are:

**Organizational Structure** events that are related to impact of organizational commitment on the agent:

- **delegate-command**: Generated when a sub-team is required to achieve a goal in the organization plan.
12.3. DETAILED DESIGN

**organization-plan-option:** Generated when the organization has more than one applicable plan that can achieve a goal.

**organization-root-option:** Generated when the intention list of a organization includes more than one ready intention with the same priority (i.e., more than one intention that can potentially execute next).

**sel-organization:** Generated when an event is posted without a responsible organization (this is just a place holder for integrating with Organization Selection but will not be implemented).

**sel-structure:** Generated when an event is posted without an adopted or committed organization structure.

**structure-option:** Generated when the organization has a number of structures it can commit to. It is used to select amongst alternative structures that are suitable for a given goal (this is just a place holder for integrating with Organization Selection but will not be implemented).

**sub-team-plan-option:** Generated when the sub-team has more than one applicable plan that can achieve a sub-goal. It is used for the organization to control the sub-team and select amongst the options for it. This event is generated both for the controlling and controlled sub-teams but with a different flag.

**sub-team-root-option:** Generated when a sub-team has been assigned two roles for the same organization, and there are two separate intentions that arise from the sub-team’s commitment to each of these roles. It is used for the organization to control the sub-team and select amongst the options for it. This event is generated both for the controlling and controlled sub-teams but with a different flag.

**goal-success-communication:** Generated when a success of a goal has occurred and the organization may be required to communicate this event to other sub-teams.

**goal-failure-communication:** Generated when a failure of a goal has occurred and the organization may be required to communicate this event to other sub-teams.

**plan-success-communication:** Generated when a success of an intention has occurred and the organization may be required to communicate this event to other sub-teams.

**plan-failure-communication:** Generated when a failure of an intention has occurred and the organization may be required to communicate this event to other sub-teams.

**Social Awareness** events that are related to the social awareness knowledge of the agent:

**new-agent:** Generated when the agent learns about the existence of a new agent.

**del-agent** Generated when the knowledge of the existence of an agent is removed.
new-organization: Generated when the agent learns about the existence of a new organization. Also generated for agents.

del-organization Generated when the knowledge of the existence of a organization is removed. Also generated for agents.

new-structure Generated when an organizational structure is added.

del-structure: Generated when an organizational structure is deleted.

add-state: Generated when new information about the state of other agents or organizations is added to the agent’s social knowledgebase.

del-state: Generated when information about the state of other agents or organizations is removed from the agent’s social knowledgebase.

All the relevant social events will be generated by the social awareness relations specified in the social knowledgebase. The organizational structure events will be generated using ephemeral relations in a specialized knowledgebase that each social agent will include.\textsuperscript{7} Thus for each such event we define an ephemeral relation in that knowledgebase.

12.3.1 Default Event Handlers

The default event handlers are implemented as dMARS\textsuperscript{TM} plans and use special utility functions for modifying the behavior of the agent appropriately. In addition we specify special command, control, and communication messages that are exchanged between the agents and are again responded to by dMARS\textsuperscript{TM} plans. In Appendix A we provide a description of the procedures (in pseudo code) to be used for each of the specified events.

Committed Organizational Structures

A committed organization is a specialization of an organization with three additional components: (1) the organization that has committed; (2) the role assignment; and (3) special access methods for querying the committed organization.

Interactions Between Committed Organizations and Intentions

The interactions between the committed organizations and the intentions are embedded in the implementation of the Perform method of the MCOrganizationStep class and the Step method of the InterpOrganizationPlan class.

The Perform method of the MCOrganizationStep class is responsible for:

1. given that the goal is to be achieved by another organization, delegating commands to sub-teams according to the command relationship defined in the committed organization and suspending the intention until the result of the attempt is reported back;

\textsuperscript{7}The only exception is the sel-structure event that will be implemented as a goal event.
2. given multiple plans that can achieve the goal, passing the plan selection decision to other sub-teams according to the control relationship defined in the committed organization and suspending the intention until the decision is reported back; and

3. reporting on the success or failure of the goal by generating the appropriate communication events.

The Step method of the InterpOrganizationPlan class is responsible for:

1. given a new intention is formed, adding it to the joint intention structure for the organization;

2. given multiple joint intentions for the organization, passing the intention selection decision to other sub-teams according to the control relationship defined in the committed organization and suspending the intention until the decision is reported back;

3. given the decision as to the selected intention, continuing execution only of selected intention;

4. given the intention has completed (successfully or unsuccessfully), removing it from the joint intention structure for the organization; and

5. reporting on the success or failure of the plan by generating the appropriate communication events.

### 12.3.2 Transformer

The basic functionality of the transformer is as follows:

1. transform the organizational structures into classes read into the dMARS™ executable.

2. transform all organization arcs (denoted by ##) into an MCOrganizationStep;

3. transform all organizational variables (denoted by &) to calls to utility functions;

4. transform every organization plan (with invocation condition with ##) into an MCOrganizationPlan; and

5. transform the required changed state communication to special databases and plans.

**Organizational Structures** are transformed into C++ classes that are read into the dMARS™ application. A organization committing to an organizational structure means creation of an instance of the structure with the appropriate role assignment. This commitment is maintained in a special list that can be accessed through special access methods. In addition, the relevant committed organization is attached to each joint intention.
Organization arcs are transformed into an MCOrganizationStep arc and the relevant logic is stored in the data member original-organization-goal. This allows the tracing of the plan to behave normally but the posting of the goal is performed on the correct logic. Furthermore the extraction of the organization goal from the arc logic is performed only once. Additional special arguments are added to the organization goal.

Organizational variables are transformed into calls to the utility function organizational-role passing it as arguments the committed organization from the invocation condition and the role name as a symbol. The function organizational-role returns the name of the organization that has been assigned the specified role.

Organization plans are transformed into an MCOrganizationPlan so the relevant methods will be called on it when necessary. Furthermore the name used for indexing is updated (to prevent all organization plans being indexed on the symbol ##) and additional special arguments are added to the invocation condition.

Changed states are communicated according to specialized database entries and plans that generate the appropriate communication events.

12.4 Analysis and Design Methodology for Organization-Oriented Systems

The field of Multi-Agent Systems has seen the emergence of a variety of approaches to the specification and implementation of agents and multi-agent systems. In recent years there has been a growing recognition of the necessity of a formal analysis and design methodology that will guide the developers of industrial multi-agent systems [22, 68, 138].

It is important to note that in general the analysis and design methodology will depend on the overall use of the system. That is, whatever the resulting system would be used to control hardware, analyse environmental conditions, or simulate social phenomena. Such systems may have very different axiom systems that would underline the produced model. As an example, a system that models emergent behavior will have the organizations and organizational structures emerge from the interactions between the various sub-teams. On the other hand, a system that performs mission critical control of a fighter aircraft may have the organizations and the organizational structures predefined and used to constrain and guide the behavior of the various sub-teams.

It is not clear if an organization-oriented analysis and design methodology could be created independent of the underlying agent model that will be used to implement the multi-agent system. This question is to some extent similar to the question of dependency between a conventional software analysis and design methodology (including object-oriented analysis and design) and the programming language that will be used to implement the resulting design [82].
Here we provide an overview of a methodology that corresponds to the operational semantics and design described in previous chapters. These in turn have been developed based on the basic organization-oriented model and axiom system. This methodology could be extended with ideas from the field of distributed object-oriented software engineering [4, 98, 99].

It is important to remember that agent-oriented systems, multi-agent systems, and organization-oriented systems are software systems. As such their development should follow the general principles of software engineering [153]. Like other software systems their life cycle would include the phases of problem analysis, system design, system development, system testing, commissioning, and maintenance.

The objective of any software analysis and design methodology is to guide the developer in a process that starts with a high-level and informal description of a required functionality and ends with a formal and detailed description of a software system that provides this functionality. This formal description can then be translated into the particular implementation system. It follows that any analysis and design methodology will depend on the characteristics and language used in the formal description of the software system.

We first provide a description of the components of a specification of an organization-oriented system. We will then proceed to provide a short description of an analysis and design methodology that will lead to such a specification.

### 12.4.1 The Components of an Organization-Oriented System

An organization-oriented system is composed of a set of organizations. The knowledge and behavior of each organization is specified through the possible social structures that the organization can adopt, the possible organizational plans that the organization can use, the type of social knowledge that the internal organizational model that the organization may have, and the mutual belief knowledgebase that the organization can reason about.

It is important to note here that both the organizational structures and the organizational plans are context sensitive means of achieving particular goals or responding to environmental conditions. The context in which such means are employed are the mental state of the organization (i.e., mutual beliefs, joint goals, and joint intentions) and the social mental attitudes it is aware of (i.e., the social relationships it has with other organizations and the social relationships that other organizations have with each other).

The organizational structures and plans specify the particular goals or environmental conditions that the organization can respectively achieve or respond to. Furthermore they specify the conditions under which the organization can achieve these goals or respond to such conditions.

### 12.4.2 The Methodology

In the methodology for designing organization-oriented systems we adopt a goal-oriented approach [32, 68]. In this approach the emphasis is on the goals that the system as a whole and its components of the system are required to achieve.
We first provide an overview of the phases in the organization-oriented analysis and design process. In general we start with a single organization that will model the whole system and then proceed to refine the design by identifying further organizations. We recursively apply the analysis and design methodology to each of the organizations identified.

The purpose of the analysis is to explore the boundaries of the problem. These include the objective of the system and the way it should behave under different circumstances. One can view this as exploring a space of possible states in which the system can be. The behavior of the system can be viewed as the paths of moving from one state to another in an attempt to reach its objective (i.e., the goal state).

One popular way of exploring this space and the possible paths through it is using *use cases* (or *scenarios*). Each use case is a particular scenario that describes a sub-set of the state space and some of the behaviors of the system (i.e., paths through this space). Our experience has shown that by carefully picking the scenarios one could obtain a very good identification of the required behavior. A similar process could be applied to sub-components of the whole system.

In the context of the organization-oriented approach the objectives of the system are referred to as the joint goals and the behavior of the system is expressed as organizational plans. These are used to form intentions that can be executed. The knowledge that an organization has about its current state is expressed as mutual beliefs. When dealing with multiple organizations one can identify the interactions between them. These are expressed as organizational structures. Let us now describe the process of analysis and design.

Each phase of the analysis and design has its own objective and we identify it with a question to be answered. We thus express the phases of the analysis and design as sequence of questions to be answered. We then proceed to describe in some detail how these steps are to be performed.

1. What are the high-level goals of the organization?
2. What declarative and procedural knowledge should be used in achieving the high-level goals?
3. Can a single organization achieve all the goals, execute all the plans, and manage all the beliefs?
4. How is the set of high-level goals, plans, and beliefs decomposed into sub-sets?
5. What organizations are to be created based on the decomposition?
6. What are the dependencies between sets of high-level goals, plans, and beliefs?
7. What are the organizational structures that should be adopted by the original organization?
8. Are the organizational structures most suitable for the goals?
9. What early decisions should be re-evaluated to allow for better organizational structures?
What are the high-level goals of the organization?

The combination of the goals achieved and conditions responded to by all the organizations in the system are a description of the required functionality. Nevertheless some goals or situations responded to may actually be required only as a consequence of some primary goal. These would be referred to as sub-goals.

Recall that we view organizations, organizational structures, and organizational plans as the means towards achieving the goals of the system. We thus need to identify each of these elements, that is, what organizations we should have in the system, what organizational structures (team and social structures) they should be able to adopt, and what organizational plans they should use to achieve their objectives.

From the perspective of a plan the basic activity units are the actions that could be taken. When developing a software system it may be the case that a basic activity unit is given by the nature of the environment or can be determined by the software designer [122]. An example of a basic activity unit that is given by the environment is a control instruction for a Commercial-Of-The-Shelf robot arm. It is clear that the various operations that are executed as part of the movement of the arm are hidden from the designer. An example of a basic activity unit that is determined by the software designer is the combination open, write, and close system calls that are part of the recording of the state of a controlled device.

In this work we will not describe how one could determine the granularity of the basic activity units. Rather we will assume that these are provided prior to the design phase. One could refer to the work of Ishida et. al. [62] for an approach to determining the granularity of an agent.

From an organizational perspective the basic organizational units are the agents. We will commence by identifying the basic organizational units in the system. An agent is defined to be an organization with no sub-teams. We start with a single organization with no sub-teams that represents the whole system. When considering an organization-oriented system one has to first answer the question: “What are the agents in the system?”

What are the agents in the system?

The approach taken to answering this question is to some extent similar to the experiments conducted by Ishida et. al. [62] in attempting to dynamically identify the boundaries of a production system (defined by the set of production rules). That is, the answer will depend on the grouping and decomposition of functional abilities. These decisions are based on the tradeoffs between:

1. The bounded ability of a single processing unit (e.g., production system, agent, operating system process, etc.) to successfully achieve all the required functionality, i.e., the high-level goals of the system; and

2. The coordination required of a group of bounded processing units when achieving their allocated goals as part of the joint achievement of the high-level goals of the
12.4. ANALYSIS AND DESIGN METHODOLOGY FOR
ORGANIZATION-ORIENTED SYSTEMS

CHAPTER 12. SYSTEM DESIGN

It seems then that to answer the first question one should understand what is required when achieving the high-level goals of the system. This primarily requires a specification of the high-level organizational plans. In addition there is also the issue of the types of mutual beliefs that are required. That is, we need to first answer the question: “What declarative and procedural knowledge should be used in achieving the high-level goals?”

What declarative and procedural knowledge should be used in achieving the high-level goals?

We will not attempt to provide here a planning mechanism or a description of general problem solving activities [93]. We will assume that the high level plans are made available either due to experience or some other method. We would like to note that the generation of the plan will depend on the specification of the basic activity units, i.e., actions.

We will note that the approach for the specification of the internal declarative and procedural knowledge of the organization is similar to that of the internal model of an agent. An example of an such an agent-oriented approach to the specification of the internal model is described by Kinny et. al. [68].

The high-level plans will include a specification of a combination of sub-goals and actions. In addition they may already include a specification of the roles that are required for executing these plans. Given these specification one can now evaluate whether these plans could be executed by a single organization. That is, we need to answer the question: “Can a single organization achieve all the goals, execute all the plans, and manage all the beliefs?”

Can a single organization achieve all the goals, execute all the plans, and manage all the beliefs?

To answer this question one has to evaluate the complexity of the plans and various knowledgebase that are required for such an activity. We will not attempt to provide here an analysis mechanism or a description of general scheduling activities [26, 47]. We will assume that such an analysis will be made available either due to experience or some other method. We would like to note that the analysis will depend on the specification of bounds on the abilities of a single processing unit.

If the answer to this question is yes then there is no need to proceed with further analysis of the organization. If the answer is no then one is required to provide a decomposition of the set of high-level goals, plans, and beliefs into sub-sets.

Decomposition of the set of high-level goals, plans, and beliefs into sub-sets

If a single organization can not execute all the plans or process all the information simultaneously then one has to divide the set of goals, plans, and beliefs into sub-sets. There are a number of ways in which such a decomposition can be done [6, 7]:

© Copyright Gil Tidhar, 1999 243
Decomposition based on the required functionality to be performed, e.g., all the activity related to interacting with external databases, all the activity related to controlling welding robot, etc. This corresponds to a function based decomposition of a human organization.

Decomposition based on the tasks to be achieved, e.g., all the activity related to the manufacturing of a chair, all the activity related to the management of a mail server, etc. This corresponds to a product based decomposition of a human organization.

Decomposition based on the physical location or resources that are required, e.g., all the activity to be performed in a particular Internet address, all the activity to be performed with a drilling machine, etc. This corresponds to a location based decomposition of a human organization.

The above decomposition will result in sets of goals to be achieved, plans to be executed, and beliefs to be managed. Based on these sets we can now create new organizations. Each such organization will be assigned the responsibility of achieving the goals, executing the plans, and managing the beliefs in one of these sets.

Dependencies between sets of high-level goals, plans, and beliefs

Given the decomposition one could now also identify the various types of dependencies between these sets. Such dependencies can be categorized similar to the approach of Lesser and his colleagues [25, 28, 47] to the analysis of task dependencies. As identified in their work such dependencies may lead to the need for coordination.

If there are no dependencies then each of the organizations can operate independently and we can repeat the same analysis process for that organization. If coordination is required then the first step is to identify the new organizations as sub-teams of the original organization. The original organization will hold the high-level plans for achieving the high-level goals. These high-level plans will ensure that the relevant temporal ordering and dependencies between the sub-goals are observed.

Given the goals to be achieved, plans to be executed (high-level and detailed), and the various dependencies we now need to answer the question: “What are the organizational structures that should be adopted by the original organization?”

What are the organizational structures that should be adopted by the original organization?

We will not attempt to provide here an organization structure design mechanism or a description of general social design activities [51]. We will assume that the organizational structures are made available either due to experience or some other method. We would like to note that the generation of the social structure will depend on the underlying communication facilities.
12.4. ANALYSIS AND DESIGN METHODOLOGY FOR ORGANIZATION-ORIENTED SYSTEMS

Note that the organizational structures for the original organization are defined by the set of sub-teams and the social relationships between the sub-teams. These structures are used to manage the coordinated activity of the organization in achieving the high-level goals of the system.

Given the organizations, organizational structures, and organizational plans one now needs to answer the question: “What internal organizational model is required by each organization in-order to operate effectively as part of the larger organization?”

Again, we will not attempt to provide here an organization model design mechanism or a description of general social knowledge design activities [14, 19, 154]. We will assume that the design of the internal organizational model is made available either due to experience or some other method. We would like to note that the generation of the internal organizational model will depend on the organizational structures and the potential social role that the organization will be required to fill.

Note that different organizational structures are means for achieving the goals. Given a particular goal a particular organizational structure may be more suitable or less suitable for achieving goals in comparison with another organizational structure. In the design process the decisions as to the particular organizational structures were constrained by earlier decisions as to the specifications of the organizations and organizational plans. At this stage one needs to answer the question: “Independent of the organizations and organizational plans, are the specified organizational structures the most suitable structures for the goals?”

Evaluation of organizational structures

The question of suitability of a particular organizational structure to a given organization, goal, and plan is one of the subjects of research of Organization Theory. This question also has to be answered when designing any distributed system. Here we do not provide an analysis of different organizational structures. Rather we note that the problem of suitability should be addressed by the analyst.

If the specified organizational structures are the most suitable structures for the goals one can proceed to refine the design of each of the organizations through a process similar to that described above. If these are not the most suitable organizational structures then one is now required to review the earlier decisions. Modifications may be required to the specifications of organizations and organizational plans to allow for the adoption of a more suitable organizational structure.

Recall that the bounded abilities of the processing units imposed constraints on the ability of an organization to simultaneously perform multiple plans or basic activity units. In a similar way organizations are limited in their abilities to command, control, and communicate with other organizations. This limitation corresponds to the bounded rationality assumption introduced by Simon [110]. These limitations force the establishment of hierarchical organizational structures.

The modifications to the specifications of organizations and organizational plans may include introduction of additional organizations in some control capacity, the development
of a hierarchical structure in which a sub-set of the identified organizations are identified as sub-teams of a newly formed organization, and more.

**Re-consideration of early decisions**

The modifications to the organizations and organizational plans may affect early decisions and one is now required to propagate the new decisions back. In particular the new decisions may affect the choice of the high-level plans. Furthermore it may affect the decomposition of the set of high-level goals, plans, and beliefs.

After this process of specification and re-consideration is completed we now have a specification of the set of organizations, organizational structures, organizational plans, internal organizational models. Furthermore each organization includes a specification of the mutual belief knowledgebase that it should include.

This specification is the high-level design of our organization-oriented system. We are now in a position to refine this design through a process that refines the design of each of the organizations and identifies organizational structures and plans for each of the organizations. It is important to note that when the process of re-consideration occurs for each of the organizations one may be required to propagate changes to the high-level design.
Chapter 13

Conclusions

Computer systems are an integral part of our modern environment. There are many types of computer systems which control various aspects of this environment. Computer systems that are embedded in the same environment affect each other through their actions. Increased networking also increases the connectivity between these computer systems. Nevertheless most computer systems do not take advantage of this connectivity and do not coordinate their activities. Nor do they make decisions that improve their overall performance and the services they provide to their users. The collection of embedded real-time computer systems that are connected to each other can be viewed as a single real-time embedded distributed system.

Our main objective was to develop a highly expressive specification language for real-time embedded distributed systems. As described in the detailed requirements in Chapter 3, we were particularly interested in a language that will allow developers and researchers to precisely specify a real-time embedded distributed system and to investigate different characteristics of such systems. We were particularly interested in a specification language that is expressive enough to cover real-time embedded distributed coordination and decision making techniques.

The main objective has been primarily addressed by providing a number of formal models of a real-time embedded distributed system. In particular we have extended known mathematical models to allow for additional features of such systems, namely, joint mental attitudes, social mental attitudes, and team structure.

As to the expressiveness of the specification language, this has been addressed at two levels. At the first level we have provided a hierarchy of models of real-time embedded systems. Each level in the hierarchy is a generalization of the previous level. Each level allows for the explicit specification of additional aspects of the distributed system. At a second level, each of the models developed included a basic model and a set of axiom systems that characterize or constrain the behavior of the system. Each level in the hierarchy allowed us to explore different aspects of a real-time embedded distributed system.

We have also proven the soundness and completeness of the models provided with respect to the associated axiom systems. We have shown that the desired characteristics specified in the semantic model would hold in the associated axiom system and vice versa.
Furthermore, we have provided a constructive method for validating formulas in each of the models.

The aspects of a real-time embedded distributed system include: (1) the components of the system; (2) the independent behavior of each component; and (3) their coordinated behavior. The details also include the division of tasks between the components, the capabilities, knowledge, and resources available to each component, the way options are evaluated and decisions made in a distributed setting, and many more.

As part of this thesis we provided the Organization-Oriented specification language. This specification language is inspired by Organization and Management Theory and allows the developer to explicitly specify all the aspects described above. Furthermore, we have provided the design of a prototype development environment that can be used for developing organization-oriented systems. We have also implemented the prototype development environment and used it to develop a particular organization-oriented system - a system for air mission modelling.

The thesis includes detailed mathematical models, system design, development approaches, and numerous examples. Let us now provide a review of the thesis and the main contribution. We also provide a short discussion about possible extensions to this work and further investigations.

13.1 Review of Thesis

In the previous chapters we introduced a hierarchy of models of distributed computer systems. This hierarchy represents a series of abstractions and enhancements. Each of the models in the hierarchy generalizes the previous model. Each of the models allows the designer of the distributed system to explicitly specify more complex behaviors of the system and its components.

At the top of the hierarchy we presented the organization-oriented model. In this model the basic component of a distributed system is an organization. An organization has a sub-team relationship with other organizations. An organization which has no sub-team relationships with other teams is referred to as an agent. The set of sub-team relationships that an organization has with other organizations is referred to as the team structure for the organization. Given that the environment is dynamic and may change over time we also allow the team structure of an organization to change over time.

We ascribe both joint mental attitudes and social mental attitudes to an organization. For any given state of the distributed system the joint mental attitudes are an explicit specification of the current beliefs the organization has about the state of the world, its currently desired states of the world, and its current commitment to bringing the desired states about. We refer to these joint mental attitudes as mutual beliefs, joint goals, and joint intentions respectively. This approach to modelling joint mental attitudes is inspired by the work of Tuomela [135].

The social mental attitudes are an explicit specification of the current relationships between the various organizations in the system. These social relationships are inspired by...
organizational models from Organization and Management Theory [7, 76, 79, 96, 103] and Command, Control, and Communication (C3) Theory [56, 64, 65, 73].

The organization-oriented model focuses on three types of relationships that exist between two organisations: (1) command; (2) control; and (3) communication. A set of social mental attitudes between the sub-teams of an organization are referred to as the a social structure for the organization. Given an organization, the combination of a team structure and a social structure is referred to as an organizational structure for the organization. An organizational structure is regarded as a means towards achieving the objectives of the organization. An organization can adopt multiple organizational structures based on its objectives and the various circumstances.

The change of focus from an agent in the system to an organization allows the designer interested in the behaviour of the system as a whole, to specify its behavior at a high level, without the need to consider all low level details. In brief, the organization-oriented model supports a form of abstraction appropriate to distributed systems.

In particular, an advantage of this approach is that in a single framework one can model distributed systems with emergent behaviour, i.e., where the behaviour of the organization is more than the sum of the behaviours of its individual sub-teams, and also model compositional behaviour, i.e., where the behaviour of the organization is merely derived from the behaviours of the sub-teams and agents.

Note that unlike previous work [55, 69, 117, 135], the organization-oriented model does not define the joint mental attitudes of the organization in terms of the joint mental attitudes of the sub-teams. Rather, it is a primitive notion for the organization. The designer can enforce the joint mental attitudes of the organization to be in particular relationships to the joint mental attitudes of the sub-teams by imposing axiomatic or semantic constraints.

Abstracting the organization’s joint mental attitudes from the sub-team’s joint mental attitudes provides a framework for investigating different team structures and their impact on the organization’s joint mental attitudes and team behavior. Furthermore, it provides a framework for investigating different types of joint mental attitudes for the organization.

Similarly, the advantage of modelling social mental attitudes explicitly for an organization is that in a single framework one can model different types of social phenomena. Examples include, “blind obedience” in which a commanded organization always adopts as its own joint goal the joint goal adopted by the commanding organization, “dominant coalition” in which the joint goals adopted by the whole organization are determined by the joint goals adopted by the commanders of the organizations, etc.

Note that in the organization-oriented model we provide an explicit representation of joint mental attitudes, social mental attitudes, and team structure. This allows the designer to specify and investigate different relationship between all three components in a single framework.

To demonstrate the advantages of the organization-oriented approach we have also designed and built a prototype development environment for building organization-oriented systems. Using this environment we have designed and built a concept demonstrator for modelling air missions. This was done jointly with the Australian Defence, Science, and
Technology Organization.

As mentioned above an organization-oriented system is at the top of the generalization hierarchy presented. An organization-oriented system in which there are no social mental attitudes between any of the organizations is referred to as a team-oriented system. The components of a team-oriented system are referred to as teams.

A team-oriented system in which all the teams are agents (i.e., they have no sub-teams) is referred to as a multi-agent system. Given that the system now only includes individual agents we refer to their joint mental attitudes as individual mental attitudes. Note that the agents can ascribe individual mental attitudes to other agents in their environment. They can use this information to communicate, facilitate coordination, or recognize beliefs, goals, and intentions of opponents.

It is important to note here that agents can have social mental attitudes with other agents. In the hierarchy of models presented here we have not explicitly described this model. Although it is interesting to investigate the different ways in which social mental attitudes could be used to affect the individual mental attitudes of an agent.

A multi-agent system in which the agents are not aware of other agents in their environment (and can not ascribe individual mental attitudes to them) is referred to as a single agent system. Note that there may be multiple agents in the system. Nevertheless they are all embedded in an environment in which they consider themselves the only agents. The model of a single agent system used here is the model of Rao and Georgeff [90] (with some minor modifications).

In a single-agent system coordination is done through the environment. It is “hard-wired” into the behavior of the individual agents. That is, it is expressed as the actions of individual agents in interacting with their (“agent-less”) environment.

To highlight the differences between the above four approaches we have presented four possible ways to specify an air mission modelling system. Each such specification was done according to one of the approaches in the generalization hierarchy.

In Table 13.1 we present the four approaches (columns) and the various characteristics of the corresponding system specification (rows). We refer to the four approaches as: Single-Agent Systems (SAS), Multi-Agent Systems (MAS), Team-Oriented Systems (TOS), and Organization-Oriented Systems (OOS). As to the various characteristics, we consider the main requirements from a model of distributed real-time embedded computer system, as described in Chapter 3. We mark with a “X” the requirements which are fulfilled by the approach.

### 13.2 Possible Extensions and Further Investigations

In this work we presented a hierarchy of models of a real-time embedded distributed system. Each level in the hierarchy included an generalization of the previous level or an additional representation of an aspect of the distributed system. Given the complexity of the models we primarily see two possible extensions to the models:

1. Enhancement to the representation of the various aspects; and
Enhancements to the Model

Enhancements to the representation of various aspects of the model include: (1) the representation of social mental attitudes; (2) representation of joint mental attitudes; (3) representation of team structure.

As to the social mental attitudes, we have adopted a Command, Control, and Communication model of an organizational structure. To some extent this can be viewed as a simplified model of Mintzberg’s [79] model of organizational structures. One could extend the model described here to include a refinement of the communication relationship as described by Mintzberg.

In addition, we have limited our discussion to formal models of organizations. In particular we were inspired by the Open Rational Systems [96] model of organizations. One can consider extending the model of organizational structures to include concepts such as power relationships, obligations, etc.

As to the joint mental attitudes, again we have considered a restricted view of the types of joint mental attitudes that can exist. In particular we have not considered emotional aspects of behavior, e.g., fear, panic, etc. These human aspects of organizations may be very important in the specification of the behavior of survivable systems.

As to the representation of team structure, our representation is limited in the aspect...
of quantitative models. That is, modelling the size of the team, the number of levels in the organization, etc. is not directly catered for. We believe that this is an area were possible extension would be most valuable.

New Combinations

In the previous chapters we only described four base models in a particular hierarchy. The various aspects of these models can be combined in different new ways to produce new models. For example, a model of a multi-agent system in which the agents have social mental attitudes can be explored.

Other combinations may include the investigation of different types of axioms system. This includes the various types of technical and social rationality. Social rationality may include the investigation of the relationships between the joint mental attitudes, social mental attitudes, and team structure and their impact on the behavior of the distributed system.

More than Models

In the previous chapters we also introduced the design and a methodology for developing organization-oriented systems. As mentioned before the design is for a prototype system and the methodology is very preliminary. Both areas could be improved and extensions could be included. Such extensions may include making the TOP system an operational system, or developing a complete organization-oriented analysis and design methodology to cover the complete life cycle of a software system, etc.

As a software development methodology there are many aspects that must be evaluated. We have only provided a very preliminary evaluation. This evaluation was constrained by the limited available data and experience and the limited scope of this work. With more use of the methodology a complete evaluation can be performed.

13.3 Main Contribution

In this thesis we presented a number of approaches for specifying distributed embedded real-time systems. In particular we focused on models of multi-agent systems. Our main criteria for evaluation of these models was the expressiveness of the specification language. The models presented differ in their expressiveness. In the more simple models only the most basic components of the distributed system can be specified. Furthermore the coordinated behavior of the distributed system as a whole is implicitly defined in the specification. In the more expressive models the developer can explicitly specify at a higher level of abstraction what are the components of the distributed system. Furthermore the developer can explicitly specify the coordinated behavior of these components. We have shown how the more expressive models generalize some aspects of the simple models.

The main contribution of this thesis included three main parts: (1) a hierarchy of models of distributed systems with increasing level of expressiveness; (2) a model of a real-time
embedded distributed system, the organization-oriented model, that has organizations as primitives and representations of joint mental attitudes, social mental attitudes, and team structure; and (3) a development environment that implements the ideas presented here and can be used to develop organization-oriented systems.
Bibliography


[31] A. Drogoul. When ants play chess (or can strategies emerge from tactical behaviors). In C. Castelfranchi and J-P Muller, editors, *Fifth European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW’93)*, Neuchatel, Switzerland, August 1993.


© Copyright Gil Tidhar, 1999 260


© Copyright Gil Tidhar, 1999 261


Appendix A

Default Event Handlers

The default event handlers are implemented as dMARS\textsuperscript{TM} plans and use special utility functions for modifying the behavior of the agent appropriately. In addition we specify special command, control, and communication messages that are exchanged between the agents and are again responded to by dMARS\textsuperscript{TM} plans. Following is a description of the procedures (in pseudo code) to be used for each of the specified events.

delegate-command

NAME: delegate-command event handler (commanding)

IC:

(delegate-command $organization
  $structure
  $commanding-subteam
  $commanded-subteam
  $command
  COMMANDING)

BODY:

(if not in command then
  (assert
   (delegated-command-attempt $organization
    $structure
    $commanding-subteam
    $commanded-subteam
    $command
    FAILURE))

(otherwise
  (send-command $organization
    $structure
    266)
APPENDIX A. DEFAULT EVENT HANDLERS

$commanding-subteam
$commanded-subteam
$command))}

NAME: command told event handler

IC:
(told
  (command $organization
   $structure
   $commanding-subteam
   $commanded-subteam
   $command))

BODY:
(assert
  (delegate-command $organization
   $structure
   $commanding-subteam
   $commanded-subteam
   $command
   COMMANDED))

NAME: delegate-command event handler (commanded)

IC:
(delegate-command $organization
   $structure
   $commanding-subteam
   $commanded-subteam
   $command
   COMMANDED)

BODY:
(post-command $organization
   $structure
   $commanding-subteam
   $commanded-subteam
   $command)
   (if SUCCESS then
(send-command-attempt $organization
$structure
$commanding-subteam
$commanded-subteam
$command
SUCCESS)

(otherwise
 (send-command-attempt $organization
$structure
$commanding-subteam
$commanded-subteam
$command
FAILURE)))

NAME: delegate-command-attempt told event handler

IC:
(told
 (delegate-command-attempt $organization
$structure
$commanding-subteam
$commanded-subteam
$command
$result)))

BODY:
(assert
 (delegate-command-attempt $organization
$structure
$commanding-subteam
$commanded-subteam
$command
$result))

organization-plan-option
NAME: organization-plan-option event handler
APPENDIX A. DEFAULT EVENT HANDLERS

IC:
(organization-plan-option $organization
 $structure
 $plans)

BODY:
(= $p (first $plans))
(assert (SelectApplicableOrganizationPlan $organization
 $plans
 $p))

organization-root-option
NAME: organization-root-option event handler

IC:
(organization-root-option $organization
 $structure
 $intentions)

BODY:
(= $i (first $intentions))
(assert (SelectApplicableOrganizationPlan $organization
 $intentions
 $i))

organization-plan-option
NAME: organization-plan-option event handler (controlled)

IC:
(organization-plan-option $subteam
 $structure
 $controlling-subteam
 $controlled-subteam
 $plans
 CONTROLLED)

BODY:
APPENDIX A. DEFAULT EVENT HANDLERS

(send-plan-control-request $subteam
   $structure
   $controlling-subteam
   $controlled-subteam
   $plans)

NAME: organization-plan-option event handler (controlling)

IC:
   (organization-plan-option $subteam
      $structure
      $controlling-subteam
      $controlled-subteam
      $plans
      CONTROLLING)

BODY:
   (= $p (first $plans))
   (assert (SelectApplicableOrganizationPlan $controlled-subteam
      $plans
      $p))

NAME: plan-control-request told event handler

IC:
   (told
      (plan-control-request $subteam
         $structure
         $controlling-subteam
         $controlled-subteam
         $plans))

BODY:
   If (got all messages from $controlled-subteam)
      (post
         (sub-team-plan-option $subteam
            $structure
            $controlling-subteam
            $plans))
APPENDIX A. DEFAULT EVENT HANDLERS

Otherwise
   (record request)

sub-team-plan-option
NAME: sub-team-plan-option event handler

IC:
   (sub-team-plan-option $subteam
    $structure
    $controlling-subteam
    $controlled-subteam
    $plans)

BODY:
   (= $p (first $plans))
   (send-plan-control-decision $subteam
    $structure
    $controlling-subteam
    $controlled-subteam
    $plans
    $p)

NAME: plan-control-decision told event handler

IC:
   (told
    (plan-control-decision $subteam
     $structure
     $controlling-subteam
     $controlled-subteam
     $plans
     $p))

BODY:
   (assert (SelectApplicableOrganizationPlan $controlled-subteam
              $plans)
organization-root-option

NAME: organization-root-option event handler (controlled)

IC:
(organization-root-option $subteam
 $structure
 $controlling-subteam
 $controlled-subteam
 $intentions
 CONTROLLED)

BODY:
(send-root-control-request $subteam
 $structure
 $controlling-subteam
 $controlled-subteam
 $intentions)

NAME: organization-root-option event handler (controlling)

IC:
(organization-root-option $subteam
 $structure
 $controlling-subteam
 $controlled-subteam
 $intentions
 CONTROLLING)

BODY:
(= $i (first $intentions))
(assert (SelectActiveOrganizationIntention $controlled-subteam
 $intentions
 $i))
NAME: root-control-request told event handler

IC:
   (told
       (root-control-request $subteam
       $structure
       $controlling-subteam
       $controlled-subteam
       $intentions))

BODY:
   If (got all messages from $controlled-subteam)
     (post
       (sub-team-plan-option $subteam
       $structure
       $controlling-subteam
       $controlled-subteam
       $intentions))
   Otherwise
     (record request)

sub-team-root-option

NAME: sub-team-root-option event handler

IC:
   (sub-team-root-option $subteam
   $structure
   $controlling-subteam
   $controlled-subteam
   $intentions)

BODY:
   (= $i (first $intentions))
   (send-root-control-decision $subteam
   $structure
   $controlled-subteam)
APPENDIX A. DEFAULT EVENT HANDLERS

$name: root-control-decision
told event handler

IC:
(told
  (root-control-decision $subteam
   $structure
   $controlled-subteam
   $controlling-subteam
   $intentions
   $i))

BODY:
(assert (SelectActiveOrganizationIntention $controlled-subteam
            $intentions
            $i))

goal-success-communication

$name: goal-success-communication
event handler

IC:
(goal-success-communication $subteam
   $structure
   $senders
   $recipients
   $goal)

BODY:
(send-goal-attempt $organization
                       $structure
                       $senders
                       $recipients
                       $goal
                       SUCCESS)
NAME: goal-attempt-communication told event handler (success)

IC:
(told
 (goal-attempt-communication $organization
   $structure
   $senders
   $recipients
   $goal
   SUCCESS))

BODY:
(assert (SetGoalAttempt $organization
   $structure
   $senders
   $goal
   SUCCESS))

goal-failure-communication

NAME: goal-failure-communication event handler

IC:
(goal-failure-communication $subteam
   $structure
   $senders
   $recipients
   $goal)

BODY:
(send-goal-attempt $organization
   $structure
   $senders
   $recipients
   $goal
   FAILURE)
APPENDIX A. DEFAULT EVENT HANDLERS

NAME: goal-attempt-communication told event handler (failure)

IC:
(told
  (goal-attempt-communication $organization
   $structure
   $senders
   $recipients
   $goal
   FAILURE))

BODY:
(assert (SetGoalAttempt $organization
  $structure
  $senders
  $goal
  FAILURE))

plan-success-communication

NAME: plan-success-communication event handler

IC:
(plan-success-communication $subteam
  $structure
  $senders
  $recipients
  $plan)

BODY:
(send-plan-attempt $organization
  $structure
  $senders
  $recipients
  $plan
  SUCCESS)
APPENDIX A. DEFAULT EVENT HANDLERS

NAME: plan-attempt-communication told event handler (success)

IC:
   (told
    (plan-attempt-communication $organization
     $structure
     $senders
     $recipients
     $plan
     SUCCESS))

BODY:
   (assert (SetPlanAttempt $organization
           $structure
           $senders
           $plan
           SUCCESS))

plan-failure-communication

NAME: plan-failure-communication event handler

IC:
   (plan-failure-communication $subteam
    $structure
    $senders
    $recipients
    $plan)

BODY:
   (send-plan-attempt $organization
                    $structure
                    $senders
                    $recipients
                    $plan
                    FAILURE)
APPENDIX A. DEFAULT EVENT HANDLERS

NAME: plan-attempt-communication told event handler (failure)

IC:
(told
 (plan-attempt-communication $organization
  $structure
  $senders
  $recipients
  $plan
  FAILURE))

BODY:
(assert (SetPlanAttempt $organization
   $structure
   $senders
   $plan
   FAILURE))

The information passed back from the controlling sub-teams to the controlled sub-teams is transferred to the relevant intentions via special ephemeral predicates.