Natural language as an Agent Communication Language

Olivia Catherine March

Honours Thesis
Department of Computer Science and Software Engineering
University of Melbourne

Supervisors
Associate Professor Steven Bird
Dr. Adrian Pearce

Abstract

Intelligent agents should be able to communicate with each other using an extensible, expressive language. Agents should have the ability work together in a heterogeneous environment to solve complex goals, while, acting on their own initiative and maintaining autonomy. Current agent communication languages are not expressive enough to facilitate coordination of agents in a heterogeneous system.

Natural languages, such as English have evolved to become expressive enough to advance the human race to be the dominant species. It has been refined over millenia and is proven extensible.

This research demonstrates the feasibility of using natural language as an agent communication language for intelligent agents solving a collaborative task.
# Contents

1 Contribution 3

2 Introduction 5

3 Dialogue Modelling 7
   3.1 The Nature of Dialogue 7
      3.1.1 Speech Acts 8
      3.1.2 H.P. Grice’s Cooperative Principle 9
   3.2 Dialogue Processing 9
      3.2.1 Symbolic vs Statistical Approach to Dialogue Processing 10
      3.2.2 Dialogue Modelling 10
      3.2.3 Tagging 10
   3.3 Dialogue Modelling Literature 11
   3.4 HCRC maptask corpus 13
   3.5 Conclusion 16

4 Agent Communication 17
   4.1 Agent Communication Languages 18
      4.1.1 Knowledge Sharing Effort 18
   4.2 KQML 19
      4.2.1 Message Format 19
   4.3 FIPA ACL 21
   4.4 Agent Communication Language Evaluation 22
   4.5 3APL 23
   4.6 Conclusion 24

5 Implementation 25
   5.1 Implementation Architecture 25
   5.2 Dialog Modelling 26
   5.3 Agent Map Representation 28
      5.3.1 Map Objects 29
      5.3.2 Object Position Representation 29
   5.4 Agents Position 29
   5.5 Agents Lexicon Representation 30
   5.6 Agent Algorithms 31
      5.6.1 Following Agent 31
      5.6.2 Instructing Agent 31
## CONTENTS

5.7 Conclusion ................................................................. 32

6 Discussion ................................................................. 34
6.1 Communication Primitives ........................................... 34
6.2 Translation Between Agent Communication Languages .......... 35
6.3 Clarification of Natural Language Ability of Agents ............... 35
6.4 English Ontology ......................................................... 36
6.5 Interoperability with Information Sources .......................... 36

7 Conclusion ................................................................. 37

8 Acknowledgement ......................................................... 39
Chapter 1

Contribution

This research has demonstrated the feasibility of using natural language as an agent communication language for intelligent agents solving a collaborative task. Natural language has not previously been reported as an agent communication for such tasks.

Extending the work of [20, 27] and [2] who stated that an agent communication languages should support communication and collaboration within heterogeneous systems. Where a heterogeneous system contains agents (possibly of different architectures), humans and information sources. The current languages produced by [20, 27] meet some of the stated criteria. The languages facilitate communication and collaboration amongst agents of a single architecture. Building on the idea that the agent communication languages should be based on language theory, namely speech acts [27, 2, 31]. We incorporated methods and theory which succeeded speech act theory in the natural language processing literature into agent communication languages. This resulted in natural language as an agent communication language.

Using natural language as an agent communication language by incorporating methods from the natural language processing literature validates the claim made by [27] in 1994. [27] claimed “applied natural language processing”, among other software technologies had “matured to the point of being ready to participate in and contribute to” programs which are distributed, heterogeneous, dynamic and which comprise “of a large number of autonomous nodes”.

As part of our research we implemented agents with English language (an instance of natural language) as their agent communication language solving a collaborative task through communication. This implementation demonstrated that agents communicating with natural language as an agent communication language is feasible.

The problems with existing agent communication languages is that they do not meet the criteria of what what their creators state an agent communication language should be [2, 12]. As previously stated, according to [20, 27] and [2] an agent communication languages should support communication and collaboration within heterogeneous systems. [12] and [2], however, identified that all existing communication languages fall short of achieving interoperability within heterogeneous systems. Using natural language as a communication language provides a language independent of an agents architecture and knowledge representation language. Our implementation allows the agents to determine the intension of received messages internally, rather than them being sent with the message as is the case with existing agent communication languages. Natural language as an agent communication language does not require an external ontology which must be shared by communication paticipants. Finally, it facilitates communication between agents and humans. These properties of natural language make it currently the best agent communication language to facilitate interoperability within
heterogeneous systems.

While our implementation demonstrated two agents communicating with natural language to solve a collaborative task there are areas which could be improved upon in future. Such as, natural language as a communication language means forming and understanding natural language utterances which are computationally more expensive than doing so using existing agent communication languages. In our implementation simple natural language processing methods were implemented, however future implementations should make use of some of the methods discussed in Chapter 3.

Natural language processing is by no means a solved problem. It is the intension of this work to demonstrate the feasibility of natural language as an agents communication language. To motivate the agent communication language community to continue to integrate developments in natural language processing in the future.
Chapter 2

Introduction

The agent paradigm for software development provides autonomous entities at design and runtime, which adapt and cooperate to solve complex tasks within distributed, heterogeneous systems [31]. For these entities (agents) to cooperate they must have the ability to communicate with other entities in the heterogeneous system. This research demonstrates the feasibility of using natural language as an agent communication language for agents collaboratively solving a task. The aim of this research is to enable agents to communicate with each other, humans and information sources alike. We propose that using natural language as an agent communication language can achieve this goal. To demonstrate, we present a case study of two agents collaborating to solve a cooperative task using English as an instance if natural language as the agent communication language.

Intelligent agents are entities which operate on a programmer’s behalf to achieve some goal or goals. An intelligent agent maintains a mental state which contains the agents current beliefs about the world. The computational behaviour of an agent can be described in terms of autonomous decisions made by the agents for the purpose of achieving its goals. An agent is autonomous if it operates without the direct intervention of humans, other agents or programmes, and has a degree of control over its own actions and internal state. It does not have control or have insight into the internal state of other agents [24]. An autonomous agent acquires its own knowledge deciding for itself “how to communicate and divide up their world”[24]. The terms autonomous agent, intelligent agent and agent are synonymous in this paper and will be used interchangeably.

Agents are increasingly being applied to distributed processing problems that require the agents to interoperate within heterogeneous, dynamic, distributed systems [27]. An agent communication language should be understandable by all entities within a heterogeneous system. In such systems agents are required to communicate and collaborate with agents built on different platforms with different architectures and humans alike [2]. An agent should also have the ability to inter-operate with information sources such as the Internet and databases [20].

Current agent communication languages are insufficient to facilitate communication with heterogeneous systems. An agent communication language is traditionally a formal language, based on English language, used by agents to communicate with one another. It is the idea of a formal language that this research will challenge.

In Chapter 3 we will examine current literature on dialogue modelling and dialogue act classification methods. Dialogue acts are used by the agents implemented in this research to provide functionality akin to speech acts in current agent communication languages. Only one method of dialogue classification is used in the implementation presented in Chapter 5, however, the literature will demonstrate the maturity of dialogue act classification. Dialogue acts are also a component in
dialogue modelling which is used as a foundation for the agent interaction algorithms and language capabilities to be explained further in Chapter 5. This chapter will provide an overview of natural language processing terminology referred to throughout the paper. Additionally, Speech act theory will be explained in this chapter as it lays an important foundation for dialogue act classification and agent programming languages. Finally this chapter will provide an overview of the Maptask corpus, where a corpus is a large collection of text and/or audio material used for natural language processing analysis and application, the plural of which is corpora. In addition, the collaborative task to be solved by the agents in this implementation is the Maptask.

Chapter 4 critically examines the literature on existing communication languages to illustrate why existing Agent communication languages are insufficient in facilitating interoperability within heterogeneous systems. Two prominent agent communication languages – KQML and FIPA ACL – will be explained in detail due to their extensive use in agent literature and implementations. Additionally, KQML and FIPA ACL exhibit properties common to other existing agent communication languages. The foundation and weakness of these languages will be highlighted, in preparation for the discussion of natural language as an agent communication languages in Chapter 6. Chapter 4 will conclude with an overview of 3APL an agent programming language used in our implementation of agents with English as their communication language.

Following the background on Dialogue modelling and agent communication languages, Chapter 5 will combine Dialogue modelling and agent theory to present our implementation of intelligent agents communicating with natural language. These agents, implemented in 3APL, work together to solve the Maptask problem using English text, an instance of natural language, as their communication language. The agent’s communication ability is modelled on the transcribed conversations of humans solving the Maptask. The agent algorithms are based on a dialogue model of the human Maptask dialogues described in Section 5.6.

Chapter 6 discusses the implementation presented in Chapter 5 in relation to the literature presented in Chapters 3 and 4. The limitations of the implementation are discussed.
Chapter 3
Dialogue Modelling

This chapter will provide an introduction to Dialogue modelling and broader aspects of natural language processing which relate to dialogue modelling. Dialogue modelling has been used for designing the agents implemented and discussed in Chapter 5. We will also introduce dialogue acts which we propose should replace speech acts for use by agents to determine the intention of a communication. This is discussed in more detail in Chapters 5 and 6.

We begin by giving a summary of speech act theory. Speech act theory provides the foundation upon which dialogue modelling, and current agent communication languages (to be discussed in the next chapter), were built.

The term Dialogue modelling will be defined in Section 3.2 in conjunction with associated linguistic terminology such as tagging. In this section we will also briefly explain the foundation of statistical and symbolic approaches to dialogue processing. These two approaches are the basis upon which the current litriture on Dialogue modelling and dialogue act classification is presented is Section 3.3.

The dialogue modelling literature is sparse, with a focus on statistical methods employed for dialogue act tagging. The literature, whilst repetitive, provides empirical evidence for the validity of using statistical methods to classify dialogue acts. The correctness of these statistical methods is approaching that of a human ability to classify utterances.

There has also been valuable work in combining statistical and symbolic methods to the field of language understanding, or rather logical representation of dialogue. This work lends itself to the implementation of English as an agent communication language. Finally we will look at the Maptask and associated corpus of transcribed dialogues. The Maptask is the problem which the agents in our implementation have been tasked with solving.

3.1 The Nature of Dialogue

Unlike monologues and texts used in natural language understanding, dialogues introduce the possibility of conflicting goals between participants (thus within a single body of text) and possible confusion during dynamic interaction.

Dialogue modelling itself is not typically a dynamic process. The process of creating the text or data to be modelled (the dialogues) is dynamic. In the scope of this paper, dialogue modelling is used in the context of extracting dialogue structure from a textbased corpus of unscripted human dialogues. Dialogue modelling will be discussed further in Section 3.2.2. First, we define some terminology and issues relating to the nature of dialogue.

At any point in a conversation each participant has an understanding what has been said. This is
CHAPTER 3. DIALOGUE MODELLING

called the **dialogue state**. A **dialogue state** is the current state of a conversation given what has passed, and should be updated after each move or utterance by a speaker. It is important for each speaker to have an understanding or maintain a belief of what the other participants in the conversation believe and know.

Throughout a conversation participants try to maintain **common ground** [23], which is essentially, an intersection of their beliefs. To maintain common ground throughout the conversation, the hearer regularly acknowledges the speaker, often through backchannels. An example of a backchannel is ’’mmm-hmmmm’. Backchannels allow the hearer to let the speaker know that they are keeping up with the conversation and that the speaker should continue.

Another factor in dialogue, which is not present in monologues and single intention texts, is turn-taking. Much work has been done on **turn-taking** in multi person dialogues, where speaker 1 says something, then speaker 2, then speaker 1 and so on.[3] Turn-taking also encompasses when a speaker should commence their next utterance. Turn-taking takes into consideration the overlap of two or more utterances and how much of a break there is between one speaker finishing and the next starting. This research is generally classified as Conversational Analysis, and is mentioned here as it has implications in Dialogue modelling. In modelling communication between humans, the silence between utterances can be long enough to take the place of an utterance, where the first utterance was intended as the first part of what can be termed an **adjacency pair**. An adjacency pair is when an utterance by speaker 1 expects a response from speaker 2. Examples of such pairs are Requests, greetings, Question-answer, etc. Taking Question-answer as an example, a long pause after the question results in speaker 1 either assuming a response or assuming the hearer did not hear the first time. This pause is called a **significant silence** as it is taken to mean something by one of the participants.

We will briefly clarify the definition of some linguistic terms to be used in the remainder of the paper which were not previously clarified. The remainder of this paper assumes the following definitions for common linguistic terms:

- **Lexicon** A speaker’s knowledge about words, also known as the speaker’s internal dictionary.
- **Lexeme** A Lexical entry, a single word and its associated information.
- **Utterance** An utterance is a complete unit of speech, bounded by the speaker’s silence. While literally referring to an auditory act, in this paper **utterance** shall be used to refer to transcribed communication or agent communication in text form.

3.1.1 Speech Acts

Speech Act theory is a pragmatic linguistic phenomenon contributed to linguistics and agent communication literature and practise by two philosophers, Austin ([1]) and Searle ([23]). Austin and Searle founded the idea that sentences or utterances are not merely sounds. Rather, an utterance is an action being performed by the speaker, analogous to a physical action, in that can change the state of the world. Austin ([1]) called such acts **Speech Acts** [3, 28]. The most common example of a speech act is a **performative sentence**, for example: [28]

I pronounce you husband and wife.

When this (or any other performative sentence) is uttered by a person of appropriate authority (such as a priest or judge in the above example), the effect of uttering the sentence changes the state of the world. An informal way of testing whether a sentence is a performative sentence is to put *I hereby*
CHAPTER 3. DIALOGUE MODELLING

... in front; only a performative sentence will “sound” correct. Speech Act theory reaches further than the performative, to encompass all possible utterances in natural speech. According to Austin there are three other classes of sentences: [3, 6]

- **Locutionary act** - the utterance of a sentence with a particular meaning.
- **Illocutionary act** - is the act of asking, promising, etc.
- **perlocutionary act** - which is the act of uttering a sentence to effect the hearer’s thoughts, feelings, etc.

Speech act theory was later refined by Searle who modified Austin’s work, by proposing that all speech acts can be better classified into five classes: Assertives, Directives, Commissives, Expressives and Declarations [3, 6]. As previously mentioned, it is the early work of Austin and Searle that led to the idea of classifying sentences or utterances and viewing them as actions.

This founded the idea for dialogue acts. A dialog act is an action which alters the state of the dialogue. Dialogue acts are discussed further in Section 3.2.2. Speech act theory has had more impact in the work on specifying agent communication languages, than it has on dialogue modelling. This will be looked at in detail in the next chapter.

3.1.2 H.P. Grice’s Cooperative Principle

H.P. Grice was another philosopher to contribute to computational pragmatics. Grice’s Cooperative Principle states that unless a speaker is being intentionally uncooperative then they adhere to four maxims, which are as follows.

Grice’s Maxims: [28, 3, 6]

- **Maxim of Quantity** (or informativeness): Say no more or less than is required by the current discourse.
- **Maxim of Quality** (or truthfulness): Do not lie or make unsupported claims, make you contribution one that is true.
- **Maxim of Relevance** (or relation): Be relevant.
- **Maxim of Manner**: Be orderly and brief, avoiding ambiguity and obscurity.

Conversational conventions such as Grice’s Maxims allow utterance meanings to be sensibly incorporated into the dialogue context and meaning. A participant in a conversation need not handle cases where there has been a violation of one of these maxims, i.e. the speaker has been deliberately misleading. The cooperative principle will form part of the assumptions in the use of English as a communication language in Chapter 5.

3.2 Dialogue Processing

Analysis of dialogue introduces interesting problems in Natural language processing. Where we would normally define dialogue as two or more participants taking turns in contributing to the conversation, for the remainder of this paper we shall assume that a dialogue is a two person dialogue unless otherwise stated.

We will now look at the foundation of the symbolic and statistical approaches to dialogue processing, and then provide a more comprehensive definition of dialogue modelling and tagging. In the next chapter these concepts will arise in the literature on dialogue modelling and dialogue act classification.
3.2.1 Symbolic vs Statistical Approach to Dialogue Processing

In the remainder of this paper we shall make reference to the symbolic and statistical (or stochastic) approach to dialogue processing. The symbolic approach to natural language processing began with Chomsky and formal language theory in the 1950's [3]. It was the initial focus of computational linguistics, working on passing algorithms and generative syntax. A symbolic approach to natural language understanding and generation is used for the English language capabilities of our agents (Chapter 5).

The statistical approach to natural language processing was seeded by statistics and electrical engineering [3]. The initial work focused on using Bayesian methods for text-recognition and authorship attribution. With the availability of the first online corpora, the Brown corpus consisting of one million words of American English texts in the 1960’s, statistical processing was used in the area of transformational grammar. Statistical methods have been applied to more areas of natural language processing as computing capacity and the amount of training data increased.

Statistical and symbolic approaches to natural language processing have existed as two separate paradigms, and still do, however, as we shall see shortly some of the literature discusses combining the two to provide a stronger approach to dialogue processing than one approach alone.

3.2.2 Dialogue Modelling

Dialogue modelling is the classification of the utterances within a dialogue using a predetermined set of dialog acts, where a dialogue act is an abstraction of the speakers intention in terms of the action type intended by the utterance [9]. A dialogue act is also referred to as a dialogue move or Illocutionary Force Type (IFT) [9] within the literature. The terms will be used interchangeably within this paper.

Dialogue acts are based on the same idea as speech acts, where an utterance is viewed as an action which effects the state of the world. However, a dialogue act is more information rich than a speech act. A speech act is intended to classify the pragmatic force of an utterance, where a dialogue act models the function that the utterance plays in the conversation [3]. A dialogue act is more than a stand alone classification it takes the context of the utterance and the dialogue state (or current state of the world) of participants into account [8].

Creating a dialogue model involves tagging each utterance with an appropriate dialogue act (or classification). The structure of the dialogue is then determined from these classifications.

3.2.3 Tagging

Tagging involves marking (or tagging) a word, or string of words with a classification label (or tag). Primarily, tagging is used in the context of Part-of-Speech (POS) tagging [3] where tagging is the process of assigning part-of-speech tag to each word. A part of speech tag is a word classification such as noun, verb, etc.

In terms of tagging in Dialogue modelling, a string of words uttered by a speaker is tagged with dialogue act or classification (core intension of the utterance) [16] such as INSTRUCTION, CLARIFICATION, etc., also referred to as a dialogue move.

Tagging can also be used to identify where utterance boundaries occur. For simple utterances, boundaries can more obviously be identified by silence. However, as previously mentioned, a silence doesn’t always mark an utterance boundary, as a pause can have its own meaning. Identifying utterance boundaries can be performed symbolically, by looking at cue words such as well, so etc. which can mark the beginning or end of utterances. [7] It can also be accomplished statistically, sentence
boundaries can be marked using POS sequences, where the POS tagger would be trained with a corpus with a sentence boundary tags set.

3.3 Dialogue Modelling Literature

In this section we will summarise and compare some of the existing approaches, the majority of which are statistical, to dialogue act classification in the literature. A statistical approach to anaphora resolution (to be defined shortly) will be presented, followed by more recent work on Dialogue Modelling which involves preprocessing of dialogues to improve the accuracy of dialogue act tagging. Finally we will examine an approach which combines both statistical and symbolic methods for extraction of meaning from dialogue.

Much of the literature on statistical dialogue act classification employs Bayesian networks. Early work in [16] examines the use of Bayesian networks to classify dialogue acts given a domain-specific, marked-up corpus of transliterate two person dialogues. These early classifications achieved an accuracy of 74% [16], with no preprocessing of the dialogue. This accuracy is comparable to that of a human whose estimated accuracy is 84% [14]. Later work by [8] similarly used Bayesian networks to classify dialogue acts in domain specific texts. This work did not extend the work of [16], differing only in the training data and dialogue act set. [8] did not provide evaluation of the accuracy of their work, nor did they acknowledge previous work on dialogue act classification [16].

In [9], probabilistic methods were similarly used to determine dialogue structure from a marked up corpus. Turn taking in addition to dialogue act sequencing were modelled using Ergodic HMMs and the ALEGRA algorithm (statistical learning methods). The main contribution of this paper was to model speaker-IFT sequencing (dialogue act sequencing), thus, little information on turn taking was presented. As with [8], [9] did not provide any accuracy measures for their work on dialogue act classification.

Progressing from simply training Bayesian nets on marked up corpora of dialogues, [13] introduced Dialog Macro-game Theory where utterances within dialogues are classified using dialog macro-grames, seemingly identical to dialog acts. Dialog Macrograms are purported to be able to be used broadly, in more diverse natural dialogues than it’s predecessors (dialogue acts). [13] provided no evidence to support the claim that this so-called new approach is any more far reaching or effective than the classification approaches mentioned previously.

From 1996 to 2002, there seems to have been little advance on the traditional training of statistical methods using marked-up corpora to tag dialogues with dialogue acts. In 2002 and 2003 new approaches to using probabilistic methods for dialogue act tagging, representing the meaning of dialogues and for anaphora resolution were presented. These three approaches to dialogue modelling propose some novel additions to existing approaches for solving traditional problems.

[14] presented an approach which provided empirical improvement of the accuracy of dialogue act tagging. [14] used discourse chunking to improve automated dialogue act tagging. Chunking, is also known as light or partial parsing, where parsing is assigning (grammatical) structure to a string or sentence. Light or partial parsing involves assigning an incomplete structure to the phrase or string. It is often used in information extraction or message understanding where a complete parse is not required [3].

Chunking was used by [14] to segment the dialogues into conversation segments. As a tag for a dialogue act is dependent on the context of the utterance, chunking allows an abstract context to be attributed to segments of the dialogue and used in the determination of the dialogue act tags thus providing richer context information for more accurate tagging.
Following the work using Bayesian nets by [16] and [8], [17] used Bayesian networks for probabilistic anaphora resolution. Anaphora is the linguistic phenomenon of referring back to an item mentioned previously in the text or conversation. [15]

\[
\text{The man has a red and a blue jumper.}
\]
\[
\text{He likes the red one best.}
\]

Using the above example sentences to explain further, the referring word (one), is the anaphora. The referring expression is said to be anaphoric. [3] The word or entity (jumper) to which it refers is the antecedent. Anaphora resolution is the process of matching, or resolving the anaphora with the correct antecedent.

While [17]’s experiments did include dialog act tagging using Bayesian networks, this research was only briefly mentioned in the paper, citing [8]’s previous work rather than referring to their own independent implementation.

Finally, [4] combines a symbolic and statistical method into a co-operative model for dialogue understanding. Firstly the dialog is parsed with a finite state model (FSM), followed by a statistical (naive bayes) learning model (SLM). Feng’s approach to discourse understanding is slightly different to the previous approaches in that rather than tagging the dialog [4] determines a semantic representation of the speakers utterances. The agents in our implementation also determine the semantic representation of utterances using symbolic methods.
3.4 HCRC maptask corpus

The HCRC Maptask corpus is a collection of audio and transcribed two person dialogs taken from adults solving the maptask.

The map task is a cooperative task solved via unscripted communication between two participants. The participants are given each given similar maps. The instruction giver is given a map with a path marked on it (see figure 3.1) and the instruction follower is given a similar map without a path marked on it (figure 3.2).

![Figure 3.1: Instruction Giver's Map](image)

Much of the audio and transcribed linguistic material available, is based on scripted dialogues or monologues. The HCRC Maptask corpus is purely unscripted human interaction. This is particularly important for the purpose of modelling natural language communication as the Maptask corpus contains naturally occurring English dialogue between two adults. Scripted material is designed to elicit a wide range of linguistic phenomenon. Scripted dialogue does not reflect spontaneous human com-
munication and thought. As previously mentioned, natural language is important for true language modelling. There is little value in modelling scripted dialogue as time has been spent cultivating the language for the purposes of the author. Naturally occurring language is more reflective of language that would occur between two humans or an agent and a human. It is the purpose of this case study to enable agents to communicate in a similar manner.

In modelling human communication for agents whose only percept is text based communication, it is important that all information communicated by the humans to solve the Maptask is modelled. For half of the Maptask dialogues participants allowed to have eye contact. This may result in facial expressions and head movements, such as nodding communicating information to the other participant, thus contributing to solving the task.

The remaining half of the dialogues, the participants had a screen between them allowing only speech based form of communication, removing all other percepts from effecting the dialogue. It is the screened dialogues that have been looked at modelling the dialogue for the foundation of the agents in this research.

The task is for the ‘instruction giver’ to instruct and explain to the ‘instruction follower’ how to follow the path in relation to the objects marked on the map. The instruction follower must mark the path on their map based on the instructions and directions given by the instruction giver. The success of the task is evaluated by visually comparing the path marked on the map by the instruction follower to the path marked on the instruction Giver’s map.

Both maps are similar, in that both maps share an overlapping set of objects, which for the most part are in the same position. However, the two maps are not identical. An object maybe missing from one of the maps, objects may have different names (for example ”Mill Wheel” and ”Old Mill”) and there may be two of a particular object on one map, and only one on the other map. Both the participants are told that the maps differ prior to attempting the task, however they are not told how they differ. The differences are left to the participants to discover and resolve through communication.

There are several different maps used in the creation of the HCRC Map Task Corpus, however only one will be discussed in this case study of agents solving the Maptask.
CHAPTER 3. DIALOGUE MODELLING

Figure 3.2: Instruction Follower’s Map
3.5 Conclusion

Much work on dialogue modelling has involved the use of Bayesian networks trained using dialogue act tagged corpora. [16] demonstrated a classification accuracy within 10% of a human’s ability to identify dialogue acts for utterances correctly is achievable.

Much of the research in dialogue act classification has been stand alone dialogue act classifiers and such. Including this research into the field of communicating agents would provide a real time test bed in practical integration of research. Additionally, once mature, transcribing conversations between agents would provide a source of training data for future statistical research.

Little work has included the use of symbolic methods for Dialogue Modelling due to the capacity of computers and an increasing availability of training data. While a statistical approach provides justifiable advantage, there is validity in symbolic approaches for Dialogue modelling. As demonstrated by [4] and [14], combining preprocessing of the dialogue whether it be symbolic or statistical and a Bayesian network trained for dialogue act classification improves accuracy. While none of the literature reviewed looked at symbolic methods alone, symbolic approach to dialogue act classification was used in the implementation of agents communicating with English to solve the Maptask. In the following chapter, we will look at the current status of agent communication languages.
Chapter 4

Agent Communication

For a multi-agent system to be successful, the individual agents within that system must be able to communicate in an effective manner \([20, 27, 2]\). That is, agents must be able to share information, knowledge and requests. The importance of agent communication cannot be understated – ineffective agent communication can lead to misinterpretation and the subsequent failure of a multi-agent system to successfully complete assigned tasks \([2]\).

The usual method of addressing communication within multi-agent systems is to establish a formal language which agents can use to communicate. Individual agents are implemented according to a common specification, communication is highly structured, and formal languages specific to the system are employed. Such formalisms, much like modern programming languages, tend to be based on English language. These formalisms are rarely compatible with one another and, in some cases, are not even compatible with different implementations of the same agent communication language. As a consequence, there are many agent communities which are unable to inter-operate with one another. \([12]\)

For heterogeneous systems, communication takes place between agents (possibly on differing platforms), humans, and external software applications. For communication in such systems to be effective, a common form of communication is required. The existing strategy for addressing this is, again, to use an English-based formalism. However in this case, the use of such formalisms can actually reduce effectiveness. If a formalism is too rigid, human users are essentially required to learn a new language. Additionally, agents communicating using another language would require a mechanism for mapping the new formalism with the existing one.

As heterogeneous agent systems are applied to more complex problems, the need for effective agent communication has led to increased attention from the agent research community \([12, 31, 27, 5, 24, 2, 30]\), building on the earlier work of \([20]\) and \([31]\). This chapter examines the current state of agent communication, its history, and the major contributors to the agent communication language literature.

We will begin by looking at the The Knowledge Sharing effort (Section 4.1.1) which introduced the concept of agent language specification to the research community. Followed by a look at the first agent language to be specified, KQML \([20]\). In addition we will look at FIPA ACL, a popular Agent Communication language which succeeded KQML.

Section 4.2 and 4.3 focus primarily on the work of three major contributors to the agent communication literature. The three prominent researchers in the field of Agent Communication Languages whose work will be reviewed are: Tim Finin, Yannis Labrou and Philip Cohen.

Tim Finin \([20, 27, 31, 32]\) was one of the founders of agent communication language specification
as co-chair of The External Interface Working Group [20, 27], one of four working groups set up by the Knowledge Sharing Effort (KSE), specified KQML[20]. Finin later teamed with Labrou [10, 31, 32] to further refine KQML and specify FIPA ACL[31]. Much of the literature on these languages has been a direct result of Finin and Labrou and will be discussed further in sections 4.2 and 4.3.

Cohen [18, 19, 2] contributed significantly to the literature of agent communication languages, providing an independent analysis of KQML and FIPA ACL. Cohen’s other significant contribution is to the theory of what properties an agent communication language should possess; This is examined in more detail later in this chapter where is provides a foundation for further discussion on the choice of English as an agent communication language.

Finally, Section 4.5 describes 3APL. 3APL the an agent programming language used in our implementation. This chapter will conclude with a comparison of literature with regard to the properties and agent language should possess.

4.1 Agent Communication Languages

Communication has shaped the face of human evolution. Complex animal families have survived by communicating experiences, dangers and information throughout generations [21]. Autonomous agents should be able to interact with other agents, software systems and humans alike through their external interface, which we shall call agent communication. The language with which these interactions occur are called agent communication languages. Traditionally an agent communication language is a formal language used by agents. Many agent communication languages (starting with KQML to be discussed presently), have been based on Austin [1] and Searle’s work [22, 23] on speech act theory which was explained Section 3.1.1.

Agents require communication language to provide an interface with the external world, to communicate and cooperate with other agents and their environment.[31] An agent communication language should provide a medium for agents to communicate independent of the agents architecture and implementation platform [20, 27, 31] thus allowing agents to maintain autonomy. For communities to exist agents must be free to inter-operate in heterogeneous communities.[27, 2] Existing agent communication languages do not sufficiently support communication within heterogeneous communities, [2] as will be discussed further in Section 4.4. KQML is one instance of a common agent communication language.

4.1.1 Knowledge Sharing Effort

The most influential paper in initialising the formalisation of agent communication and knowledge representation was [20], which documented the culmination of the Knowledge Sharing Effort (KSE). The Knowledge Sharing (KSE) effort was established in 1991 by Defence Advanced Research Projects Agency (DARPA). Its purpose was to develop a formalism and technical structure to enable the sharing of knowledge between knowledge base systems, humans and software systems [20].

The KSE set up four main task forces to address the four impediments to knowledge sharing reported in [20]: Heterogeneous (knowledge) Representations, Dialects within (knowledge representation) Language Families, Lack of Communication Conventions and Model Mismatches at the knowledge level. The main focus of the KSE was to develop formalisms to support sharing, preserving and build on existing agents. These formalisms centred around agent communication languages and ontology specification. [20]
Our main interest in the remainder of this paper will be the third impediment (Lack of Communication Conventions) for which the External Interfaces Working Group was set up.

The purpose of the External Interfaces Working Group (EIWG) was to define a set of communication (query and response) protocols for knowledge based systems.

EWIG was faced with the task of specifying a protocol for knowledge bases (or agents) to interact with one another and other information sources such as databases “in a run-time environment” [27]. Finin [27, 20, 31] began his contribution to agent communication literature as co-chair of the External Interfaces Working Group. The main product of EWIG was Knowledge Query and Manipulation Language (KQML) which encompasses a specification of a communication protocol and language.

KQML is one of the two agent communication languages focused on in this chapter due to its proliferation in the agent community and it being the first specified ACL [27]. KQML also had a significant effect on later agent communication languages, specifically FIPA ACL. Additionally, KQML in comparison to FIPA ACL is a convenient vehicle for illustrating the problems with existing communication. These problems can be addressed by using natural language as an agent communication language.

In the following section KQML will be explained in detail and we will further explore the contribution Tim Finin has made to Agent Communication Language specification.

4.2 KQML

KQML was the first attempt at the formalisation of an agent query and response protocol. KQML was intended to be for agents what SQL had become for databases [12], for querying and responding to queries by other agents and software systems.

KQML loosely consists of a protocol containing some mandatory and optional fields. Looking at KQML from an ACL perspective, it is the performative and message content of KQML message passing that is of primary interest.

KQML is based on the theory of speech acts - the idea that language consists of performatives which describe an utterance that changes the world in the same way as a physical action alters the state of the world. These performatives or intentions are not always clear cut for a given utterance[10]. An utterance can be literal, non-literal, fit a structured or template form or not. According to [31], communication can only be successful if the receiver(s) determine the intention or intended meaning of the message content. Consequently, KQML allows a message to be sent only if the associated performative (intention behind the communication) is included as a mandatory field of the protocol. The reason for this structure is that, given the performative of an utterance in conjunction with the utterance, the intension of the utterance is unambiguous.

4.2.1 Message Format

KQML has three layers, the first of which is the content layer. The content layer contains the body of the message, the format and language of which is dependent on the communicating agent implementation. The content language is independent of KQML and can include the Knowledge Interchange Format (KIF) specified by the KSE, first order logic, English or some language known only to a specific implementation of agents. The figure below is an example KQML message which uses LPROLOG as it’s content language.

The middle layer of KQML is the communication layer, which encodes message features that describe communication parameters. These parameters include the unique identifier of the commu-
CHAPTER 4. AGENT COMMUNICATION

20

communication and the identity of the sender and receiver(s). In the case of the figure the communication
identifier is \textit{ibm-stock} and the sender and receiver are \textit{joe} and \textit{stock-server} respectively.

Finally the message layer determines the network protocol by which the message is delivered and
attaches the performative specified by the sender to the content [31]. The performative of the example
is \textit{tell}.

Example of a KQML message([31]):

\begin{verbatim}
(tell
 :sender    joe
 :content   (PRICE IBM ?price)
 :receiver  stock-server
 :reply-with ibm-stock
 :language LPROLOG
 :ontology NYSE-TICKS
)
\end{verbatim}

KQML is purported to be extensible by allowing the addition of new performatives at anytime. In
order to facilitate extensibility, no formalisation of the semantics of performatives [2] was published
in the initial KQML specification. This lead to agents of different implementations having different
semantic interpretations for the performatives. Thus, several different KQML dialects have been
created, resulting in KQML agent communities being unable to interoperate with one another[12].

For correctness, technically KQML has never been implemented due to the fact that it is a proto-
col. Agents that communicate using KQML have been implemented. This holds for all agent
communication languages discussed throughout this paper. The phrase \textit{implementation of an agent
communication language}, means an agent communicating with the specified language has been
implemented.

Finin [20, 27, 31] claimed that KQML was implementation independent, and that there existed
several implementations in different programming languages and different agent architectures able to
interoperate [27]. The success of the interoperability cited by Finin is due to a single organisation
being responsible for the implementation and specification of the agents A single organisation that
shared a single semantic specification of the performatives.

KQML was designed to be an implementation independent, extensible language, in which not
specifying the performatives allowed for the addition of new performatives at any time [27]. KQML
can only be implementation independent for agents that share the same semantic specification for the
performatives. Different semantic specification results in different interpretation of performatives by
communicating agents, thus, KQML is not universally implementation independent. Consequently,
KQML is extendible, however, it is not extensible. New performatives can be added to the language,
however, agents implemented before these performatives were add have no mechanism for interpreting
the semantics of the new performatives.

Tim Finin, in conjunction with Yannis Labrou refined the specification for KQML in 1997, to
include preconditions, postconditions and completion conditions for the performatives. An example
of the KQML semantic specification of the \textit{tell} performative from the previous KQML example [31].

\begin{verbatim}
tell(A,B,X)
Pre(A): Bel(A,X) && Know(A, Want(B,Know(B,S)))
Pre(B): Int(B, Know(B,S))
\end{verbatim}

where S may be any of Bel(B,X), or \( \neg (\text{Bel}(B,X)) \)
CHAPTER 4. AGENT COMMUNICATION

Post(A):
\[
\text{Know(A,Know(B,Bel(A,X)))}
\]

Post(B):
\[
\text{Know(B,Bel(A,X))}
\]

Completion:
\[
\text{Know(B,Bel(A,X))}
\]

The precondition specifies the necessary state of the sending and receiving agents. Conversely, postconditions specify the states of the sending and receiving agents after the performative has been sent and accepted, prior to the next performative. \cite{31} The completion condition differs, as it specifies the state after a conversation and the initiating performative (and the intention behind it), is fulfilled \cite{31}.

While not intending to tie the language to any specific architecture, the semantic description provided is in terms of the agents mental attitudes with relation to its belief, desires, knowledge and intentions \cite{10} thus binding the new version of KQML to the well known Belief, Desire, Intention (BDI) model. \cite{12}

KQML was the first specified agent communication language. It was intended to be an extensible, implementation independent language. The lack of initial semantic definition led to multiple dialects of KQML and this led to agent communities which were unable to interoperate. The addition of semantic description did not benefit agents already implemented, nor does it support implementation independence, now being suitable for BDI agents only. FIPA ACL overcame some of these problems.

4.3 FIPA ACL

FIPA ACL is a protocol based agent communication language, superficially similar to KQML. FIPA ACL is identical to KQML syntactically, however semantically they are quite different. In this section we will describe FIPA ACL and the impact of KQML on development of FIPA ACL.

Yannis Labrou \cite{27,31} was a founding member of the Foundation for Intelligent Physical Agents (FIPA) academy and has contributed significantly to the literature on agent communication languages. Founded in 1999, FIPA is an international organisation focused on the development of standards in the broad area of agents. Labrou actively worked on the specification development team for FIPA ACL \cite{31}.

As previously mentioned FIPA ACL is syntactically identical to KQML. It’s reserved words, called Communicative Acts, differ from those reserved for KQML’s performatives. Both Communication acts and Performatives are analogous to a communication acts in the respective languages.

FIPA ACL message \cite{29}:

\[
\text{(inform}
\begin{align*}
:sender & \text{ agent1} \\
:receiver & \text{ agent2} \\
:content & \text{ (price good2 150)} \\
:language & \text{ sl} \\
:ontology & \text{ hpl-auction}
\end{align*}
\text{)}
\]

This example message is from agent1 to agent2, of type (communication act) inform. The language of the content message in the above example is SL, a language specific to FIPA ACL. Note that while SL is used in this message, the content message language, can be anything as with KQML.
Widespread criticism of KQML for lack of formal semantics led to FIPA ACL defining its own semantic formalism [29, 2, 31]. FIPA ACLs formal language for defining semantics is called SL. SL is based on modal and multimodal operators which describe the precondition and effect of each Communication act. It is a requirement that agents communicating using FIPA ACL have some understanding of SL. SL is closely tied to BDI with the Communicative acts described in terms of an agents Beliefs, Desires and Intentions, etc.[31] Thus, FIPA ACL is tied to the BDI agent architecture, limiting its ability to promote interoperability between heterogeneous agents. [2]

The Communicative acts of FIPA ACL are specified in terms of the acts’ feasibility precondition and rational effect. The feasibility precondition, specifies the necessary condition of the sending agent. The rational effect describes the expected result of the Communicative Act in terms of the receiving agent. An example of semantics for the communicative act inform (used in the example FIPA ACL message) [29]:

\[
< i, \text{inform}(j, a) > \\
\text{feasibility precondition: } B(i)a \land \neg B(i)(Bf(i)a \lor Ub f(j)a) \\
\text{rational effect: } B(j)a.
\]

Where,

\( B(i)a \) means agent \( i \) believes \( a \);

\( Bf(i)a \) means agent \( i \) is definite about \( a \); and

\( Ub f(i)a \) means agent \( i \) is uncertain about \( a \).

As the example messages show, FIPA ACL and KQML share identical message syntax and neither are committed to a content language (the language the message is actually written in). Despite being both based on Speech Act Theory [31], KQML and FIPA ACL use different names for, and different semantic description of, their communicative primitives[31]. There is no direct mapping between KQML and FIPA ACL communication primitives, thus one cannot translate between FIPA ACL and KQML.

4.4 Agent Communication Language Evaluation

In 1999 Finin and Labrou [31] surveyed the “Current Landscape” of Agent Communication Languages, consisting solely of their respective work on KQML and FIPA ACL. Labrou and Finin provided a comparison of the two languages on the basis of the differences between the languages, rather than their strengths and weaknesses. Providing an informative overview of the two languages with a clear bias to the validity of these languages as agent communication languages. Finin and Labouru conceded many KQML based applications are unable to interoperate due to the lack of agreed upon semantic specification prior to the refinement of KQML. However, Finin and Labouru later claim the lack of interoperability of KQML implementations is due mainly to a “lack of real motivation” [31] by the implementors. They also claimed that the lack of formal semantics is “much less important than it sounds” [31] and that the weakness in agent communication is not whether or not the agents can speak the same language, it is that there is no central registry for agents to allow agents to send and receive messages. This view is not shared by [25], [2] and [12]. The latter of which is a more comprehensive and objective paper on the current status of agent communication languages. [12] critically evaluated six Agent communication languages, the language descriptions in this paper contained errors in fine detail. For example, the content language of KQML is KIF (Knowledge interchange format, which is also a product of the KSE) [12]. The Knowledge Interchange Format (KIF) is based on a first order logic and was for the purpose of being a common language for agents to express properties of a set domain [29]. As explained in Section 4.2 KQML message content can be in any language specified
CHAPTER 4. AGENT COMMUNICATION

by the communicating agents, such as prolog, lisp, English and KIF.

[12] did identify several flaws in current agent communication languages. In addition to highlighting limitations of agent models which in turn place limitations on the language ability of agents. The work presented in [12] will now be summarised to give a broader perspective on the weaknesses of current agent communication languages.

The first weakness of these languages is lack of formal semantics exhibited by languages such as Agent Oriented Programming (AOP) and KQML. The lack of semantic definition has resulted in a lack of consistency across implementations.

AOP is an agent model based on Object orientated programming (OOP) and speech act theory. AOP agents are limited to three communicative acts, inform, request and unrequest, which, like the initial specification of KQML.

The second weakness is, insufficient available performatives and a lack of extensibility of performatives. Which AOP and Mobile Agent Communication (MAC) (which limits inter-agent communication to requests) both suffer from. This weakness restrict the level of complexity at which such agents can interoperate.

FIPA ACL while providing a large selection of semantically specified performatives (Communication Acts). FIPA ACL is described as “not robust enough for mission critical systems” due to it’s close ties to the BDI architecture by [12]. Other agent communication languages tied to a specific agent architecture include refined KQML (tied to BDI), Inter Agent Communication Language (ICL) of the Open Agent Architecture (OAA), and AOP (which is tied to OAA). These languages do not promote interoperability within heterogeneous systems. The goal of agent communication languages are to facilitate communication and coordination between agents, software systems and humans alike. As such, agent communication language tied to a specific architecture cannot effectively serve a heterogeneous community.

4.5 3APL

Triple APL (An Abstract Agent Programming language) [26] provides a testbed for intelligent agents. A 3APL agent comprises of a belief base, goal(s), capabilities (rules for updating beliefs) and reasoning rules (through which an agent’s goals can be revised).

The transition algorithm built into 3APL applies the reasoning rules to the goals in the agent’s goal base. Additionally the message transport system for communication between 3APL agents is built into 3APL. The agent’s belief base, capabilities, goals and rules are specified by the user. The belief base contains the agent’s knowledge about the world. The agent’s beliefs can be updated via its capabilities.

Capabilities consist of a precondition, a capability or action, and a post condition:

\[
\{ \text{Pre} \} \quad \text{Act} \quad \{ \text{Post} \}
\]

The precondition (Pre) must be satisfied before the action (Act) can occur. Following the action, the post condition (Post) specifies the updates made to the agents belief base. The belief base contains facts about the state of the world. Updating the belief base consists of adding and removing facts.

The Goal base specifies the goals of the agent. The agent continually operates on its goals (as specified by the reasoning rules), resolving them where possible, until the goal base is empty. An agent can have two types of goals: a basic goal and a complex goal. The basic goal can be a basic action, test goal or a predicate goal [26]. The reasoning rules capability of the agents allows the agent
to revise or remove blocked basic actions and unachievable goals, optimise the agent’s goals and be used to define predicate goals [26].

3APL is available in both Java and Haskell implementations. The Java implementation (unlike the Haskell implementation) has a GUI interface, conforms strictly to the FIPA ACL standard for inter agent communication and has extensive prolog support [11]. The Haskell implementation was developed for prototyping. It has a number of features not yet available in the Java implementation: Support for multiple (concurrent) goals; multiple, repeated goal variables in rule body and patterns; most general goal selection (and capabilities for the user to specify the goal selection algorithm) and support for external components in any language which can read from and write to standard I/O.

The Haskell implementation of 3APL has been chosen for implementation of an agents system with English as the agent communication language. The Haskell implementation allows a foreign language interface for the natural language processing modules (to be discussed shortly) and does not require the use of FIPA ACL for communication between agents.

4.6 Conclusion

An agent communication language should facilitate interoperability between agents, humans and software systems alike. The language should be independent of the agent’s architecture or implementation platform thus supporting heterogeneity. [2, 12] It should be extensible rather than merely extendible. [24, 2] Extensions to the language should be recognisable by earlier implementations (the language additions should be backwards compatible), expanding as new needs arise. [24] An agent communication language should also support scalability, allowing the number of agents within a multi-agent system to increase without negative effect on the communication language. Language conventions should be adaptive to allow new meanings to enter at anytime [24]. An agent communication language should not require stringent ontologies; Rather participating agents should be able to evolve their communication abilities to allow for additions to the language.[24] An agent communication language should allow for an open community in which agents may join at any time [2, 24].

The major efforts in agent programming language specification, have been just that. Producing a formalism to define interactions between agents and the domain in which they operate. Despite the literature promoting heterogeneity and interoperability, agent communication languages proposed to date do not facilitate these properties within agent systems. KQML attempted to achieve these goals by leaving the definition of its performatives open to enable extensibility. The lack of formal definition resulted in several KQML dialects that could not interoperate. FIPA ACL, syntactically similar to KQML and semantically more stringent, resulted in different implementations able to interoperate but at the cost of constraining implementations to a BDI architecture. In the literature examined in this chapter there currently does not exist an agent communication language that supports communication within heterogeneous systems. In the following chapter we shall look at an agent implementation which uses English as an agent communication language.
Chapter 5

Implementation

An agent communication language should enable agents to communicate within a heterogeneous system. The agent communication languages as described in Chapter 4 do not achieve this goal. In this chapter we propose English language, an instance of natural language, as an agent communication language for the purpose of agents solving a collaborative task through communication and describe an agent implementation which uses English language as the communication language. Our agents are implemented using the Haskell implementation of 3APL (Section 4.5) to solve the Maptask problem (Section 3.4).

The Haskell implementation of 3APL has been chosen for implementation of an agents system with English as the agent communication language. The Haskell implementation allows a foreign language interface for natural language processing modules (which are external to the agent and will be discussed shortly) and does not require the use of FIPA ACL for communication between agents.

We will begin by describing the architecture of the agent system, the way in which the human dialogues from the Maptask corpus were modelled to determine the structure of the agent interaction.

The agent’s internal representation of the Maptask problem then explained, and finally we describe the algorithm used by the rule based agents to solve the Maptask.

5.1 Implementation Architecture

The objective of our implementation is to demonstrate agents communicating with English language (as an instance of Natural Language) to solve the Maptask problem.

The implementation consists of two agents: an instruction follower agent and an instruction giver agent. Each has it’s own natural language processing module, implemented in python. This is shown in Figure 5.1. Each agent has different abilities, with abilities being determined by the agent’s role in the maptask. Extending both agents to carry out both roles, requires that each agent include the other’s capabilities and resolution rules. Thus the agents take on either role by simply changing their goal.

Both NLP modules use a rule based (symbolic) approach to convert the agent’s utterance from it’s knowledge representation to English. Upon response of an utterance, pattern matching is used to translate the English language communication received into the agent’s knowledge representation language.

Unlike the agent’s internal representation, the two natural language modules use different English grammars. This is intentional, using different grammars is intended to be more realistic and to avoid the situation where the speaker converts it’s communication to English using pattern matching, and
Figure 5.1: Agent Implementation Model This figure shows the modules which make up the agent system in this implementation. It shows the flow of information between the modules and the language in which this information is written.

The hearer translates the English communication into its representation language using the same set of pattern matching rules. The pattern matching rules are extracted from modelling the human dialog of people solving the Maptask problem, which we shall now discussed.

5.2 Dialog Modelling

The agent algorithm is designed by modelling the dialogue of humans participating in solving the Maptask. In designing this algorithm, transcripts of dialogs between humans were analysed by classifying each human utterance. The classification classes are Statement, Agreement, Instruction, Clarification and Correction. A sample of a categorised dialog is shown in Figure 5.1. The sample is part of a categorised transcript of two humans solving the Maptask using the same maps as used by the agents in this case study (Figures 3.1 and 3.2). The human utterances in the Maptask corpus are more complex than the agent dialog. The agent dialogue follows the simplified format of Instruction, Clarification and Agreement. The agents employ basic turn-taking rules for communication, with agent 1 speaking first, then agent 2 and so on. The utterance boundary is defined as the end of the communication.

One agent forms a knowledge representation sentence of what it wants to communicate to the other agent. The sentence is passed into the natural language module where it is translated into English. The English sentence is passed back to the agent, which then sends it to the hearer agent. The Natural language sentence that the agent uttered is stored in the speaker’s knowledge base.

The receiving agent sends the natural language sentence to its Natural Language Module, which translates from English to the agents’ knowledge representation. The knowledge representation sentence is passed back to the agent which interprets it, and updates its knowledge base depending on the content of the communication and the agent’s capabilities. The translated sentence and the Natural Language sentence that was initially received is also stored in the knowledge base. This enables the agent to “roll back” its state in the event that the current (or prior translation) was incorrect. (ie. the utterance was misinterpreted).

In the event that the natural language module requires more information to translate the current
Table 5.1: **Classified Human Dialogue** This table contains a sample of the classified human dialogue from the Maptask corpus

<table>
<thead>
<tr>
<th>Person</th>
<th>Utterance</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Giver</td>
<td>Okay, starting off we are above a caravan park.</td>
<td>STATEMENT</td>
</tr>
<tr>
<td>Follower</td>
<td>Mmhmm</td>
<td>AGREEMENT</td>
</tr>
<tr>
<td>Giver</td>
<td>We are going to go due south straight south. And then we’re going to turn</td>
<td>INSTRUCTION</td>
</tr>
<tr>
<td></td>
<td>straight t back round and head north past an old mill on the right-hand</td>
<td></td>
</tr>
<tr>
<td></td>
<td>side.</td>
<td></td>
</tr>
<tr>
<td>Follower</td>
<td>Due south and then back up again?</td>
<td>CLARIFICATION</td>
</tr>
<tr>
<td>Giver</td>
<td>Yeah, south and then straight back up again with an old mill on the right</td>
<td>REITERATE INSTRUCTION</td>
</tr>
<tr>
<td></td>
<td>and you’re e going to pass on the left-hand side of the mill.</td>
<td></td>
</tr>
<tr>
<td>Follower</td>
<td>Right, okay</td>
<td>AGREEMENT</td>
</tr>
<tr>
<td>Giver</td>
<td>Okay? And then we’re going to turn east.</td>
<td>INSTRUCTION</td>
</tr>
<tr>
<td>Follower</td>
<td>Mmhmm</td>
<td>AGREEMENT</td>
</tr>
<tr>
<td>Giver</td>
<td>not straight east, slightly sort of northeast</td>
<td>CORRECTION</td>
</tr>
<tr>
<td>Follower</td>
<td>Slightly northeast</td>
<td>CLARIFICATION</td>
</tr>
<tr>
<td>Giver</td>
<td>very slightly. And we’re going to continue straight along. quite a wee</td>
<td>REITERATE and</td>
</tr>
<tr>
<td></td>
<td>distance. Right we’re gonna continue along on that course and then we’re</td>
<td>INSTRUCT</td>
</tr>
<tr>
<td></td>
<td>going to turn north again.</td>
<td></td>
</tr>
<tr>
<td>Follower</td>
<td>Mmhmm</td>
<td>AGREEMENT</td>
</tr>
<tr>
<td>Giver</td>
<td>And well a distance below that turning point there’s a fenced meadow.</td>
<td>CLARIFICATION</td>
</tr>
<tr>
<td></td>
<td>But you should ld be avoiding that by quite a distance.</td>
<td></td>
</tr>
<tr>
<td>Follower</td>
<td>Okay</td>
<td>AGREEMENT</td>
</tr>
</tbody>
</table>
utterance, the natural language module can query the knowledge base for the required information. For instance, if the sentence being translated contains a referring expression, the NLP module can query the agent for information required to resolve the referent. This query may consist of a request for previously uttered and received NLP sentences or a request for information from the agent’s belief base, such as the speaking agent’s current location (as believed by the hearer agent).

5.3 Agent Map Representation

The agents in this implementation have percepts limited to text based communication. Thus the map and the associated path must be represented in each agent’s knowledge base so as to facilitate text based conversation and allow updates. The agents are rule based agents and their knowledge is represented as a set of prolog facts. Therefore the Maptask map is also represented the map as a set of prolog facts.

The following diagram (Figure 5.2) contains a part of the instruction Giver’s map. Example 1 shows how the highlighted section of the map is represented in the agent’s knowledge representation language.

![Diagram of Instruction Giver's map with highlighted path](image)

**Figure 5.2: Path of Instruction Giver Agent** This figure show the Instruction Giver’s map with a segment of the path highlighted. Example 1 shows this highlighted section in the agents knowledge representation language, as it appears in the agent’s belief base.

```prolog
path(rel_pos(n, caravanPark), rel_pos(w, pickettFence)).
path(rel_pos(w, pickettFence), rel_pos(s, pickettFence)).
```
Example 1: This is the agents way of representing the segment of the path highlighted in Figure 5.2.
As the path closely follows the objects on the map, the path is represented in the agent as a series of path segments. The segments consist of a start and end position of the agent, relative to an object on the map. As shown in Figure ?? the fact with which the path is represented (path/2) contains two relative position relations. The relative position relation (rel_pos/2) consists of a compass direction and an object which the direction is in relation to.

5.3.1 Map Objects
Each of the objects on the map are words (or lexems) in the agent’s lexicon. The objects’ name and a list of identifying features and associated words are stored in a prolog fact, called lexeme/2.

eg. lexeme(forest, [trees, grass, dense]).

The agent is able to update the knowledge pertaining to the objects in the lexicon in addition to having the ability to add new words and synonyms which will be discussed in Section 5.5.

5.3.2 Object Position Representation
The position of an object on the map is defined in relation to the other objects on the map. The position rule (pos/2) consist of an object, as a single name, and a list of relative positions of the object with respect to other objects on the map. The relative positions take the form of discrete compass coordinates, eg N, W, NW.

eg. pos( millWheel, [ rel_pos(ne picketFence), rel_pos(se, caravanPark), rel_pos(w, fencedMedow)] ).

This form of representation is selected for two reasons. Firstly the maps of the Maptask problem are coordinate free (see Figure 3.1), thus a coordinate based system would be inappropriate. Secondly the maps are scale free pictorial maps, there are no frames of reference other than the objects on the map.

5.4 Agents Position
An agent’s position is represented in relation to an object. An agent may be on any side of an object or at one of the corners of the object, described as a compass coordinate relation. An agent cannot move in a given direction from the object it is at if the object is obstructing the path the agent wants to travel. In this case the agent must go around the object to the side it wants to leave from.

Figure 5.3 shows the the compass directions in relation to the object. Figure 5.4 shows the agent on the south side of the object. The way this position would be represented in the agent’s knowledge representation is shown in Example 2.

curr_pos( rel_pos( s, abandonCottage ) ).

Example 2: This fact is how the agent represents being in the position shown in Figure 5.4.
5.5 Agents Lexicon Representation

The agent represents each object on the map as a propositional fact that contains the name and a list of properties of the object.

As the agent’s English language capability is contained within the natural language processing module attached to the agent, an understandable assumption is the lexicon of objects would be stored there too. However, the agent needs to know the definitions of an object within the knowledge base is to facilitate the agents ability to resolve synonyms and update the lexicon with new objects. The lexicon describes all the objects on the map thus is tied to the representation of the map which is stored in the knowledge base of the agent. Adding a new word to the lexicon corresponds to adding a new object to the map.

If a word uttered by the speaker is unknown by the hearer, the hearer may request an explanation if it believes the word important. If the speaker knows the word it will offer a description of the word. If this description matches that of another word in the hearer’s lexicon, then the hearer believes the new word is a synonym of the existing word. The new word is then added to the hearer’s belief base. The hearer tells the speaker about the synonym (word already in the hearer’s belief base), the speaker will then add the synonym to it’s belief base. Over time this means that both agents will develop a common lexicon.
CHAPTER 5. IMPLEMENTATION

If an agent learns a word or fact from another agent, such as the location of a previously unknown object on the map, it will add the new knowledge to it’s belief base. The acquired knowledge will be noted as being from another agent in case the information or situation in which it is given is misinterpreted and the information is incorrect. The agent is able to revert to a prior state if it knows what information is incorrect.

5.6 Agent Algorithms

The agents implemented in this investigation resolve their own utterances, through clarification and query communications. They do not violate the constraints of being an autonomous agent, by examining or specifying each other’s ontology, prior to communication as with KQML (Section 4.2). Rather, they make use of their inbuilt communication ability to resolve conflicts in their ontology. As discussed in the previous section the agents ability to resolve unknown words, which can be used to resolve any word, whether it be an object, an instruction, or the position of an object. The agents can query each other until they have resolved the word or conflict, or resolution is not possible. This enables the agents to communicate with a wider pool of agents and humans for whom their ontological representation differs from that of the agents implemented here, so long as entities participating in the communication can speak English.

As previously discussed the agents are rule based agents where resolution rules dictate what action the agent takes next. In Section 5.1 the interaction between the agents and their natural language modules was explained. In the following two sections the algorithms of the agents and their interaction with each other, will be explained.

5.6.1 Following Agent

The design of the Following Agent’s algorithm is shown in Figure 5.5. The main aspect of this algorithm is that it waits for an instruction from the Instruction giver Agent. Once received (and translated from English to the knowledge representation by the natural language module), the agent matches the instruction to it’s knowledge base (interprets the instruction) identifying each word. In this instance the words are either a direction (n = North, nne = North, North East) or an object or frame of reference, such as Caravan Park (caravanPark) or Abandoned Cottage (abandonedCottage). If any of the words are unknown, the Following agent asks for an explanation, to which the Instructing agent can give one of two replies. 1. It doesn’t know the word, in which instance the Following agent assumes the natural language module has misinterpreted the communication. The Follower agent will ask for a clarification of the utterance. If the clarification utterance from the Instruction giver cannot be resolved the information which can be extracted is used. The utterance is added to the Followers belief base so it can be fully resolved sometime in the future. 2. Alternatively the instruction giver knows the word and offers a description. If this description matches that of another word in the Follower’s lexicon, then the agent believes this other word is a synonym of the new word, adds the new word into the belief base and tells the Instructing agent about the synonym. If the utterance is an instruction to move the Following agent finally checks whether it knows the location to the object it is moving to. If it does, then the agent moves and send an AGREEMENT message message to the Instructing agent to let it know that the Follower has followed the instruction. It then begins to wait for the next instruction. If the location of the destination is unknown, the the agent asks the Instructing agent where the object is.

5.6.2 Instructing Agent

The Instruction Giver agent’s algorithm is somewhat simpler (see Figure 5.6) than that of the Instruction follower. The Instruction giver issues an instruction to move (from where the Instructing agent believes the Following agent to be) to the next point along the path. The point on the path is in terms of relative position between the agent and a landmark on the Instructing agent’s map. The Instructing agent then proceeds to wait for an AGREEMENT message from the following agent, indicating that the Following agent has heard, understood and followed the instruction. However, the Following agent can clarify either words contained within the message or the location of the object if it hasn’t heard the word before or doesn’t have the object on it’s map respectively.
CHAPTER 5. IMPLEMENTATION

32

Figure 5.5: Instruction Follower Algorithm This diagram shows the algorithm of the Instruction Follower agent

5.7 Conclusion

In this chapter, an implementation of agents using English as a communication language in 3APL to solve the Maptask was described. Part of the Maptask corpus was used to model human dialogue upon which the agent algorithms were based. The transcribed dialogues were also used to extract rules upon which the natural language ability of the agents was based. The symbolic approach used in the agent algorithms was a result of using 3APL a rule based agent programming language. A symbolic approach to the agent’s natural language ability could be improved using statistical classification of dialogue acts for more efficient translation into the agents knowledge representation format. Statistical methods were not employed in this research due to a limited amount of training data. This and other aspects of the implementation will be discussed further in the next chapter.
Figure 5.6: **Instruction Giver Algorithm** *This diagram shows the algorithm of the Instruction Giver agent*
Chapter 6

Discussion

In this paper the term communication was used as meaning the act of allowing the exchange of thoughts, ideas, and information through speech or written media. As discussed in Chapter 4, an agent communication language (ACL) should enable communication within a heterogeneous system while promoting autonomy. A heterogeneous system may contain agents possibly of differing implementations and architectures, humans and information sources such as the Internet and databases. In this chapter we shall discuss how existing ACLs are insufficient in enabling communication within a heterogeneous system while promoting autonomy and why English language as an instance of natural language is a suitable agent communication language.

Languages such as KQML and FIPA ACL take the form of a communication protocol. The intention of the message is communicated as a field within the protocol rather than being communicated in the content message. The content languages of the messages in these protocols are unspecified to allow for flexibility. This intended flexibility has led to the agents communicating with languages such as KQML and FIPA ACL to be isolated from other agents who speak the same agent communication language, but use different content languages.

6.1 Communication Primitives

In 1994 [27] stated that “applied natural language processing”, among other software technologies had “matured to the point of being ready to participate in and contribute to” programs which are distributed, heterogeneous, dynamic and which comprise “of a large number of autonomous nodes”.

The natural language processing community have been modelling human speech since the work of Searle and Austin. The later work, however, has made little impact on the agent communication language literature. The agent communication language research community continues to rely primarily on speech act theory founded by Searle and Austin. The classification of speech acts such as performatives, has provided a convenient foundation for creating formal agent communication languages. This convenience has prevented the agent communication language community from integrating new communication theory into agent communication languages. The agent communication language literature has not made reference to any natural language processing methods or theory since the work done by Searle and Austin. Speech acts aid in deciphering the intension of the message content. However, there has been natural language processing work on utterance classification primarily using statistical methods, since the 1960’s, as discussed in Chapter 4.

In specifying performatives as part of the communication language, one simply introduces another barrier for heterogeneity. KQML and FIPA ACL illustrate this barrier, in having different names and semantic definition of their communication primitive. KQML and FIPA ACL do not enable mapping from communication act to primitive thus preventing automated translating. In the implementation presented in Chapter 4 the agents used simple symbolic methods for extracting the dialogue act from the utterance.

The term dialogue act is distinct from speech act. Dialogue acts take into account the dialogue state or the leading dialog as effecting the tag associated with an utterance. As discussed in Chapter 3, dialogue act tagging has received significant focus in the dialogue modelling literature. Employing more sophisticated dialogue classification techniques is outside the scope of this investigation could be used in future. By relying on the
agents in our implementation to classify the utterances, the agents did not have the problems associated with FIPA ACL and KQML. They did not require a common semantic definition of the dialogue acts as it was an element of their knowledge base. Additionally, in not specifying the intent of the communication, the agents did not make explicit assumptions about the knowledge representation of other agents.

The definition of an autonomous agent is to hide the internal workings of the agent. It should not support remote method invocation, rather a communication should enable intensional exchange of information without reference to the architecture or implementation detail of the agent. In addition to employing the use of communication primitives the definition of these primitives is tied to the BDI architecture for both Refined KQML and FIPA ACL. In a heterogeneous system agents should be able to communicate regardless of their architectures. KQML and FIPA ACL do not support communication within heterogeneous systems. Using natural language as a communication language enables agents and people to communicate without concern for the knowledge representation of fellow communication participants. The knowledge representation is irrelevant to the communication as message disambiguation and intention identification are performed solely within the agent.

6.2 Translation Between Agent Communication Languages

The KSE identified that there should be a way of translating from one agent communication language to another automatically [20]. As explored in Chapter 4 the more popular ACLs are currently based on English, namely speech act theory and are presently not compatible with one another. Even FIPA ACL and KQML who share syntactic similarities are unable to interoperate due to different classification terminology and semantic definition for communication primitives. The implementation in this paper has used a dialogue act set to classify utterances as a precursor for message understanding. This was done inside the agent’s natural language processing module rather than sent by the speaker agent as with KQML and FIPA ACL. Therefore, the agents were not required to use the same set of dialogue acts.

KQML and FIPA ACL performatives do not have a direct mapping to one another. Using English as a natural language, translating English messages can be done using similar NLP modules as used by the agents in this implementation. This would allow translation from English to KQML and vice versa. Training data tagged with the communication primitives of the languages which the English is being translated into would be required for creating a dialogue model. The dialogue model would be used for translation, where the performatives would be considered the dialogue act tags for this model.

Allowing different systems with different mechanisms to choose a classification for interpretation of meaning of an utterance allows agents and humans to interact without requiring conversation participants to have identical performative definitions. In addition to this, humans rarely consciously classify what they hear. Thus eliminating the speech act element of the communication language in favour of English enables communication between agent systems and humans to occur more freely than using a strict protocol such as KQML.

6.3 Clarification of Natural Language Ability of Agents

In implementing agents with natural language capability it is not proposed that an agent should possess human level intelligence. In our implementation in Chapter 5, the agents communicating with natural language were domain specific. The agents need to be able to communicate with English, be able to expand their lexicon and describe words. Depending on the computational ability and architecture of a natural language agent, it may use a symbolic or statistical approach to Natural language understanding. As mentioned in Chapter 3 for natural language understanding or knowledge representation purposes, it is not necessary to understand an entire utterance, rather enough understanding to extract meaning is sufficient. Thus the agents in our implementation used symbolic natural language interpretation as best suited a rule based agent.

In addition to the internal natural language capabilities of an agent, external lexicons such as wordnet, are available for agents with online connections to enable expansion of language understanding capabilities. New words can also be added to an agents internal lexicon, this through the agents ability to question other agents involved in the conversation.
CHAPTER 6. DISCUSSION

6.4 English Ontology

The literature proposed that for agents to be successful they require identical ontologies [20, 31]. [20] claimed that to build agents more efficiently required a central bank of agent ontologies which could be reused. KQML and FIPA ACL specify the agent ontology as a field of their protocol. Communicating agents using these languages must have identical ontologies in order to communicate. This requires the agents to have the same knowledge representation. Natural languages can be viewed as an ontology. For example, if I show you a picture of a tree, you too will think it is a tree. The natural language capability suggests the agents internally define objects and words within the chosen natural language. Therefore there is no need to specify an ontology when the agents are communicating with natural language.

In using English as a content language for a current agent communication language such as KQML. There is nothing preventing the agent ontology of these agents from mapping English words to different meanings to those we as humans know. For example, the word tree may be mapped to an image of a dog within the ontology.

In early papers such as that of [20] central storage of data such as ontologies and other agents components could reduce the time and effort costs in building new agents. Agents communicating with natural language can use resources such as wordnet to expand their language capability. Online resources such as wordnet are larger than any agent specific resource or ontology bank. Wordnet allows agents and people alike to look up words, definitions etc.

6.5 Interoperability with Information Sources

The final property that we stated an agent communication language should possess is that it should enable communication with information sources such as the Internet and databases.

Communicating with information sources such as databases is as inefficient as communicating with other agents using a formal language. For formal agent communication languages, there must be a mapping from the agent query language to that of the database or information source for query and response to be feasible.

Many information sources on the Internet, for example as webpages, contain natural language from which information can be obtained by a natural language speaking agent. Unlike dialogue, webpages can not be queried when a new word or concept is encountered. Definitions need to be determined by the agent either through reference to such tools as wordnet or querying other agents in the system. An agent with English capabilities is more likely to be able to obtain information from a webpage than an agent who communicates using FIPA ACL.

In querying databases English language capability does not offer any obvious advantages over existing ACLs such as KQML or FIPA ACL. However, most database query languages provide online manuals written in natural language from which mappings between English and the query languages might be determined. This would be more feasible than for KQML or FIPA ACL agents communicating in these languages require a mapping between their content language and the database query language. As there are an unlimited number of content languages, creating a mapping for all content languages and ontologies for each communication primitive is inefficient. Additionally, a database query is unlikely to return an intention or performative, therefore, translation of the information returned by the database would be required for all content languages and ontologys.

We have discussed how using natural language as a communication language is architecture independent. In addition, natural language promotes autonomy by leaving the disambiguation of intention to the hearer thus allowing cross compatibility interactions between agents and humans and in doing so facilitates translation between agent languages. Natural language communication has the capability to provide an agent ontology. We looked at agent communication languages interacting with information sources which is an aspect of interoperability in heterogeneous systems where natural language provides only a slight improvement of existing agent communication languages.
Chapter 7

Conclusion

This research has demonstrated the feasibility of using natural language as an agent communication language for intelligent agents solving a collaborative task. Natural language has not previously been reported as an agent communication for such tasks.

We built on the work of [27] and [2, 24]. [27] claimed “applied natural language processing”, among other software technologies had “matured to the point of being ready to participate in and contribute to” programs which are distributed, heterogeneous, dynamic and which comprise of a “large number of autonomous nodes”. [2, 24] stated that an agent communication language should facilitate interoperability within heterogeneous systems.

Employing natural language, specifically dialogue modelling methods, we have implemented an agent system using English (an instance of natural language) as a communication language to solve a collaborative task through communication.

The work by the Knowledge Sharing Effort [20] in 1991 stated for agent programming to move forward, ways to “preserve existing knowledge bases and of sharing, reusing and building on them” must be found [20]. The languages specified since are formalisms which required communication participants to have the same terminology for communication primitives (speech acts), the same semantic representation of these primitives, identical ontologies and the same content languages in order to communicate.

The common property of formal communication languages is that each utterance has a label, a communication primitive or classification of intention of the message being sent. The communication primitive is derived from the work of Austin and Searle on Speech act theory. The work of [2] and [12] shows that these communication primitives introduce a point of incompatibility. Employing dialogue act classification methods from the natural language enables agents to classify messages themselves, thus, removes the need for agents to conform to a single set of communication primitives with a static semantic description. Our implementation used simple symbolic methods for utterance classification, understanding and generation. The literature reviewed in Chapter 3 provided several other methods which could be employed by agents for utterance classification and understanding. Further work on employing dialogue act classification and associated natural language processing methods would provide agents with more complex natural language capabilities, expanding their ability to interoperate with other agents, humans and information sources.

[21] stated that Intelligent agents maintain a state of the world and have the purpose of achieving one or more goals. In using speech acts the agents are viewing the messages as stand alone messages, similar to method calls in mainstream programming. [31] asserted that agents do not support remote procedure calling and method invocation. Their communications are more expressive. This research discovered that an utterance by an agent should be viewed as a dialogue act rather than a speech act.

Speech acts are independent classifications utterances which do not take into account the context of the utterance. Based on the work of [24], an intelligent agent maintains a state of the world which it updates given its perception of changes that occur and ones that it effects. Communication affects the state of the agents world. Communication should therefore be viewed as a state, a dialogue state that is maintained and updated given new utterances by dialogue participants. Speech acts are not intended for this purpose. This functionality is a property of dialogue acts.
It is the belief of the author that the formalism of agent communication languages needs to be shed for agents to effectively communicate and solve tasks in heterogeneous systems of the future. Speech act theory, while an important foundation for dialogue modelling and agent communication languages, is by no means the final word in natural language processing. We believe that speech act theory should not be the final contribution to the agent communication languages from the perspective of human language. Human language has evolved over millennia; employed by agents it provides a flexible extensible language which enables agents and humans to communicate with one another.

Our work provides a foundation for integration of natural language processing techniques and agent communication languages. This work demonstrates the feasibility of natural language as an agent communication language. The agent communication language community needs to continue to integrate developments in natural language processing in the future.
Chapter 8

Acknowledgement

I would like to thank the following for making this paper possible:

**Mike Ciavarella** for the wisdom support, encouragement, insanity and laughing at my spelling. Apologies for making you read and understand intelligent agents.

**Steven Bird** for his patience, support and pushing me to achieve things I otherwise would not have attempted.

**Adrian Pearce** for his continual encouragement, time and understanding.

**Ioanna, Chris and mpp** for their friendship, continual support, company online late at night and invaluable procrastination coaching.

**Yoshi** for technical support, RAM, selflessness and friendship I wouldn’t have survived without.

**Claire** for the hilarity, insanity, insults and being the best office mate a person could have.

**Chris Leckie** for patience, support and encouragement.

**Adam Hendrix (ASP)** for friendship, laughter, torment and employment.

**MUCSA and company** for pup-nights and putting up with me.

**Les Kitchen** for putting up with me.

**Tariq, Scott, Larry, Suresh and Vanessa** for the violence and company in first semester.

**Fellow honour students** for Company, friendship and laughter.

**Albert** for keeping my work safe, never crashing, speed and providing hours of entertainment.

**dictionary.com** for the words ispell doesn’t know.

**Family** for your love and understanding.

**Bert** for your love, support, inspiration and drive which taught me never stop trying, my thoughts are with you always, R.I.P.

**YOU** for getting this far - thank you for reading my work

Thank you to everyone who has supported me throughout my undergraduate degree. Thank you again to my supervisors Adrian Pearce and Steven Bird for their time, ideas, patience, invaluable support and putting up with me!
Bibliography


