Emotion-led modelling for people-oriented requirements engineering: the case study of emergency systems

Tim Miller, University of Melbourne
Sonja Pedell, Swinburne University of Technology
Antonio A. Lopez-Lorca, Swinburne University of Technology
Antonette Mendoza, University of Melbourne
Leon Sterling, Swinburne University of Technology
Alen Keirnan, Swinburne University of Technology

Abstract—In the field of design, it is accepted that users’ perceptions of systems are influenced by emotion as much as cognition, and functionally-complete products will not be adopted if they do not appeal to emotions. While software engineering methodologies have matured to handle non-functional requirements such as usability, what has not been investigated fully is the emotional needs of people. That is, what do users want to feel, and how do they feel about a system? In this paper, we argue that these emotional desires should be treated as first-class citizens in software engineering methodology, and present preliminary work on including emotions in requirements models using emotional goals. We evaluate these models both with a controlled user study, and on a case study of emergency systems for older people. The results of the controlled user study indicate that people are comfortable interpreting and modifying our models, and view the inclusion of emotions as first-class entities as a positive step in software engineering. The results of our case study indicate that current emergency systems fail to address the emotional needs their users, leading to low adoption and low usage. We conceptualised, designed, and prototyped an improved emergency system, and placed it into the homes of nine older people over a period of approximately two weeks each, showing improved user satisfaction over existing systems.

1 INTRODUCTION

“...even if a design is elegant and functional, it will not have a place in our lives unless it can appeal at a deeper level, to our emotions.” — Hartmut Esslinger [46, p. 9].

Evidence suggests that inadequate consideration of requirements are a major cause of software project failure [12]. In the context of technology adoption, users reject a technology or use it in limited ways when their needs and experiences with that technology are not addressed [23, 24]. As the famous quote from Esslinger above conveys, this is especially true in the case of social objectives such as the emotional needs of users. The consideration of emotion in addition to cognition has become more prevalent in design in recent years [30], including human-computer interaction design, but such considerations have not transferred successfully to software engineering, despite evidence showing that a user’s acceptance of product is typically based on emotion rather than cognitive [30]. This is especially important in domestic or social systems, in which workflows are loose and people do not generally have the well-defined roles and responsibilities found in organisational settings.

Software engineers are trained to build systems with desired functionality and non-functional properties. However, software systems are often designed poorly, detracting from the user experience. Cooper [8] refers to this as “the inmates running the asylum”: software engineers elicit functional and non-functional requirements from users, then design a product to fulfill these requirements as they themselves would like it to be, resulting in software that fails to fulfill the desires of its intended users. This problem is made worse by a common misconception that problems with the interaction design can be addressed after the development by simply fixing up the user interface.

From the perspective of software engineering, a first important step of addressing user experience is eliciting the emotional desires of stakeholders. A growing appreciation can be found in literature that existing software engineering methods are limited by not considering social objectives [2, 39, 51], a view expressed well by Baxter and Sommerville [2, p. 14] in their comprehensive review of design methods for socio-technical systems: “Modelling and abstraction is fundamental to software engineering, with models of different types being used by engineers to communicate. The practical use of socio-technical approaches has to acknowledge this by providing a means of modelling, and by integrating with existing approaches. [...] The abstractions currently used in technical system modelling (e.g., use-cases, objects, etc.) do not seem to us to be sufficient to represent socio-technical considerations.”

In previous work [27, 36], some of the authors presented a systematic and repeatable process and method for understanding the roles and goals within a social domain for the purpose of informing technology design. At the heart of the method were agent-oriented models [43]. Ethnographic data was collected using a variety of means, and analysed using a grounded analysis. We used agent-oriented models to record the ground theory

1. By “design” here, we refer to the design of the product, not the software architecture or detailed designs.
that resulted from that analysis. An important aim of the work was to provide a simple yet flexible modelling notation that could be used to create boundary objects, which, as Paay et al. [32] demonstrate, can be used as shared artifacts between stakeholders from different disciplines. We designed and implemented technology probes [17], and modelled the data collected from these probes as agent-oriented models. These models allowed us to represent human activities as well as software system behaviour. One novel outcome was the use of quality goals to represent socially-oriented requirements such as “having fun” and “being playful”.

In this paper, we improve our previous work by adding the concept of emotional goals [21] to the notation and method, which capture the desired feelings of stakeholders in a socio-technical system, and how these relate to the system and each other. We call these models people-oriented software engineering (POSE) models, because of their focus on the people within the system, as well as the software.

In this paper, we present a two-part evaluation of our models. First, we present a user study in which we compare our notation for capturing user needs against the well-known social modelling notation i* [54]. We asked a set of participants, some technical and some non-technical, to answer a series of questions about an i* model and a POSE model, and measured the time and accuracy of their responses. We then asked a set of qualitative questions around their preferences between the models. The results show that participants understand POSE models better and more quickly, prefer POSE models as boundary objects for modelling socio-technical systems, and prefer the use of explicit emotional goals in models over our previous approach of using quality goals to represent these social aspects.

Second, we evaluate the concept of emotional goals via a case study on emergency alarm systems. These systems allow a person to raise an alarm in the case of an emergency, and also to “check in” (wellbeing check) each day to convey that they are well by pressing a button. We interviewed 12 participants about emergency systems and their feelings towards technology in general. Using the ethnographic data collected, we modelled the emotional, functional, and quality goals of the key stakeholders using our models. Based on the findings of the case study, we discovered that many users of emergency systems, as well as their families, were not happy with the way the system operates for them. While the technological systems themselves were well-engineered, reliable products that fulfilled the functionality of alarms and wellbeing checks, the emotional needs of users were not met, leading to failure of the overall goals of the system. That is, older people did not receive assistance when needed because they were not carrying their devices, and did not feel that the wellbeing check supported their wellbeing. In the process, we learnt lessons about our models, and how to improve them. From the resulting models, we designed and built a new prototype of an emergency system. Evaluation of this prototype demonstrated an improved user experience over existing emergency alarms.

We next present our argument as to why emotions should be embedded as first-class citizens in software engineering, and in Section 3, we present relevant background for the paper. In Section 4, we present our modification to our previous modelling notation to include emotional goals. In Sections 5 and 6 discuss our evaluations of this using a user study and the case study of emergency alarm systems.

2 Emotions as First-Class Citizens in Software Engineering

In this section, we outline our argument for why emotions should be considered as first-class citizens in software engineering methodology.

The consideration of emotions in requirements engineering is not new, but has received insufficient attention outside of the games community, with only a handful of papers addressing the issue of how to address emotions in software engineering [3, 6, 7, 40, 47]. Further, as far as the authors are aware, integrating emotions fully within requirements engineering has not been explored, nor has carrying emotions through the software engineering lifecycle. As part of our larger research program, we aim to model emotions as first-class entities in software engineering, carrying these goals through the software engineering process, including requirements engineering, product design, software design, implementation, testing, and validation.

2.1 The case for emotions

The idea of eliciting emotional desires for product design is not new, and there has been a large body of work over the previous two decades; see Desmet and Hekkert’s editorial for the special issue Design & Emotion in the International Journal of Design [10] for an excellent overview of this work.

Norman’s book on emotional design [30] is one of the most seminal pieces of work on this topic. He argues that designers must elicit desired user emotions and explicitly address them as part of the design process. Norman describes how three levels of the human brain affect emotion, and what this means for designers:

1) Visceral processing is the automatic, pre-conscious processing that makes fast judgements. Visceral processing is programmed in humans, meaning that its effect is fairly consistent across different people. With regards to design, a person’s emotional state is affected by visceral processing based on the appearance or “look & feel” of a product, such as colours and style.

2) Behavioural processing is sub-conscious, and is the part that controls “everyday” behaviour. With regards to design, behavioural processing is about
the use and experience with a product, with the experience related to “function, performance, and usability” [30]. Like the visceral level, emotional responses to the same event at this level are quite consistent across different people.

3) Reflective processing is conscious, and is the contemplative part of processing. It is only at this level that “the highest levels of feeling, emotions, and cognition reside” [30]. With regards to design, it is about the meaning of a product and its use. Of significance is that the reflect level extends much longer over time than the visceral or behaviour levels, which are typically about “now”. Unlike the previous two layers, emotional responses to the same event at this level often differ significantly between individuals.

The reflective level can enhance or inhibit the behavioural level, and the behavioural level can in turn enhance or inhibit the visceral level. Activity can be initiated “bottom-up” from the visceral level, or “top-down” from the reflective level, and any activity has both a cognitive component, which assigns meaning, and an emotional or affective component, which assigns value. The emotional state influences the way in which people think, including cognitive thought.

Norman provides a simple model that maps the three levels of processing to product characteristics (see Table 1). He emphasises that visceral design is all about immediate impact, and that physical characteristics such as look, feel, and sound dominate. It can be measured by showing products to people and observing their reactions. Behavioural design is more about use, and appearance does not matter. Behaviour design must consider functionality, understandability, usability, and performance, and can be measured quite objectively using metrics. Reflective design is about culture, meaning, and self image. Measuring such properties is intrinsically more difficult than the visceral and behavioural levels, partly because metrics for reflectiveness are hard to define, and partly because reflective processing happens over much longer time periods than visceral or behavioural processing. Norman emphasises, however, that reflective processing often determines a person’s overall impression of a product, so is perhaps the most important in determining whether people actually use or recommend a product.

<table>
<thead>
<tr>
<th>Level</th>
<th>Product characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visceral design</td>
<td>Appearance</td>
</tr>
<tr>
<td>Behaviour design</td>
<td>The pleasure and effectiveness of use</td>
</tr>
<tr>
<td>Reflective design</td>
<td>Self-image, personal satisfaction, memories</td>
</tr>
</tbody>
</table>

2.2 The case for emotions as first-class citizens in software engineering

To relate Norman’s three levels to software engineering, we consider the methods used to achieve them. Good visceral designs are achieved using aesthetically appealing interfaces between user and software; in most cases, they are of the form of graphical user interfaces.

Good behavioural designs are achieved by implementing useful functionality, and making products usable. These are the common traits that are considered in software engineering, and are typically divided into functional and non-functional categories, with usability, performance, reliability, etc., falling into the latter category. These non-functional properties are often termed qualities, quality goals, or soft goals.

Good reflective design is achieved with software that appeals to cultural perceptions and messages, and meaning. It is our view that modern software engineering methodology fails in providing systematic and repeatable support for reflective design.

We propose that properties desired by people at a reflective level should be termed emotional desires or emotional goals. While related terms such as value and motivation could be used here, we use “emotion” because our experience suggests that people relate the term “emotion” to these subjective and hard-to-measure properties, and because it is more clearly tied to people rather than software.

Based on Norman’s theories and the discussion above, we make two observations:

1) Emotional goals or desires are not just another form of usability. Usability is a behavioural-level experience, not a reflective level experience. While the usability of a product can influence the reflective level, as described by Norman’s theory, it extends well beyond immediate use and into long-term reflection.

2) Emotional goals or desires are not just qualities. Similar to item 1 above, qualities are behavioural, not reflective. Pragmatically, people relate qualities of a system to the visceral and behavioural aspects, not to culture and meaning, so describing the influence a software system has on the reflective level as a quality is unsuitable, as it conflates behavioural and reflective levels. For example, consider the well-known Facebook application². A goal of Facebook is to make people feel part of their friendship group. Is it reasonable to label this a quality goal, when its success depends on users’ establishing social relationships with friends and participating in meaningful interaction? A single person using Facebook will not feel part of their friendship group. Such a goal can be triggered or helped by the application, but is only achieved for specific groups of people over a specific period of time. This property must be interpreted from a social perspective, as distinct from the application’s functionality and usability.

2. See https://www.facebook.com/.
Further to this, as noted by Hudlicka [13], emotions have scales of intensity, which change over time. Assigning an intensity to a property such as usability is meaningless. The intensity of an emotion experienced during an interaction with a system is a property of the person, not of the system, even though a system can contribute to this.

Several authors have made the case for emotions in requirements engineering [3, 5, 6, 7, 40, 47, 20], and this work is discussed in the related work (Section 3.1). However, such work has had minimal impact. Maiden [20] argues that the reason requirements people do not take emotion seriously is because they are Spocks, in a reference to the emotion-free Vulcan character from Star Trek, rather than Kirks, the passionate captain. Requirements engineering methodologies tend to focus on the software, rather than the people. While the last couple of decades have seen some move away from this, such as social modelling [54] and user stories, which focus on why users want something in addition to what they want, the focus is still very much on the behavioural why, rather than the reflective why.

Further to this, we propose that emotions need to be considered further than just requirements engineering – they must be traceable throughout the entire software engineering process.

Emotional goals are abstract and subjective, so cannot be implemented directly. We propose that emotional goals are not software requirements, but merely drivers to elicit new requirements, either functional or non-functional. For example, one of our current projects is on a web application to support people with psychosis. An important emotional goal is for the supported person to feel normal. Such a goal cannot be implemented directly, because it is a property of a person, not of the system. In this project the feeling of being normal is conveyed by the contents of videos in which people recovering from psychosis narrate their experiences. The emotional goal feel normal provoked the addition of the functional requirement of using people with lived experience in the videos.

In addition, unlike other qualities such as performance and usability, we are not aware of metrics that can be applied to the system to evaluate whether it makes a person feel normal, other than asking users directly. While usability evaluations can be used to measure user perception, the longer-term nature of reflective processing means that such evaluations must be improved to evaluate whether emotional goals have been met.

We propose that emotions should be embedded as first class citizens of people-oriented software engineering methodology for two reasons:

1) It is clear that the abstract, subjective, and difficult-to-measure nature of emotional goals make them challenging. It is our view that by not explicitly considering emotions throughout the entire software engineering process, important emotional considerations risk being sidelined. They can be ignored or missed at the requirements stage, but their challenging nature means that under constrained deadlines, they will be the first things discarded in development if they are not explicitly considered – yet they are often the most important.

2) Stakeholders are unlikely to communicate emotional requirements without explicit prompts, as this is not their view of software as a cold, hard calculating machine. In our experience, stakeholders focus on functional requirements, sidelining ‘soft’ issues such as socially-oriented requirements, including emotions. Yet, their perception of any system will depend heavily on their emotional reaction to the system [30].

While we do not claim this view is required for every software system, we believe that systems embedded in social or domestic domains, and systems with loosely-defined workflow, will benefit from this most.

3 BACKGROUND AND RELATED WORK

In this section, we present related work on emotions in requirements engineering and agent-oriented requirements engineering.

3.1 Related work

3.1.1 Emotions in requirements engineering

While the role of emotions is prevalent in the field of human-computer interaction (e.g. [1, 48]), it has been scarcely explored in the software engineering literature, despite its importance being identified by prominent researchers in the field (e.g. [20, 47]). Existing work mostly considers emotions in relation to requirements engineering. In some cases, the emphasis is on acknowledging that emotions can significantly impact the requirements engineering process; for instance Thew and Sutcliffe [44, 45, 47]. They identify a number of values — referred to as soft issues — that can have different effects on requirements. They distinguish between those that are critical for the success of the project from those that have a lesser impact if not treated. Similarly, Proynova et al. [38] use personal values to identify features that are then mapped to requirements that may have otherwise remained uncovered. In both cases, the authors use personal values to prioritise or discover requirements. For us, however, emotions have a definite effect in the requirements, and personal values, in turn, determine how different people react to the same emotion.

In the requirements modelling notation i*, Yu [54] uses softgoals to model non-functional requirements. Besides classic non-functional requirements such as “reliable” or “secure”, other less traditional factors (better aligned with emotional issues) are also included, such as “trustworthy”, “flexible”, “minimal intrusion” or “normal lifestyle”.

Colomo-Palacios et al. [6, 7] explicitly ask stakeholders about their feelings regarding particular requirements.
Their response is recorded in a grid which represents in the X axis the arousal caused by the requirement and in the Y axis the pleasure. Comparing the responses with the evolution of the requirements throughout the requirement engineering process iterations, they conclude that high levels of pleasure and low levels of arousal seem to indicate accepted requirements. Although Colomo-Palacios et al. provide an interesting insight on the correlation between pleasure and arousal and requirements stability, this approach measures stakeholders’ emotional perceptions of requirements around emotional-related qualities such as pleasure and arousal, whereas we aim to use emotions to explore and define requirements with stakeholders, inline with Maiden’s argument [20].

In this sense, the work of Ramos et al. [41] is better aligned with our approach. They present examples of projects where requirements were affected by emotions, values or beliefs, and argue that the impact of these factors, traditionally ignored, is as important as functionality or qualities. They advocate identifying these issues as soon as possible and then tackling them with some managerial or technical treatment, and suggest a number of psychological techniques to identify emotional issues when talking to the stakeholders. Once identified, the requirements engineering process continues as per usual. We build on this work to elaborate on the definition of emotions relevant for the development process and to define a notation to graphically represent emotions in the context of the requirements engineering process.

### 3.1.2 Agent-oriented requirements engineering

With the agent paradigm increasingly becoming a popular and successful way for modelling complex systems [29], agent-oriented requirements engineering has been a focus of much research. Several methodologies for agent-oriented software engineering have been proposed, such as Tropos [4], Prometheus [33], Gaia [55], INGENIAS [35], and ROADMAP [43]. These different methodologies offer differing processes, methods, and notations for using the agent paradigm to engineer a complete system. However, the methodologies have several modelling concepts in common, including the use of agents or roles to capture the idea of people and systems interacting, the use of goals instead of functional requirements to capture desired behaviour, and the use of quality goals to capture non-functional requirements. The concepts are used because it is believed that they offer a natural way for describing software systems as well as the larger socio-technical system in which they operate.

In our earlier work [27, 36], we have investigated software engineering for people-oriented systems. In particular, we have focused on methods and processes for identifying the functional goals of users, the qualities of these functional goals, the roles they play in systems, and how these goals are translated from requirements to design, implementation, and validation. We used the agent-oriented paradigm for high-level modelling because we are particularly interested in capturing the goals and needs of the users and stakeholders of systems, and not just functional and non-functional requirements of the software. We used the concepts of roles, functional goals, and quality goals to capture users, what they want to achieve, and how they want this to be achieved; and also to capture software requirements.

From the perspective of requirements engineering, we identify stakeholders of the system, which are modelled as roles, and elicit information that answers the following two questions: (1) what do you want to achieve (functional goals)? and (2) how should it be achieved (quality goals)? For example, a potential user may want to be able to share pictures to their friends (functionality), and they want this to be fast and easy, hinting at non-functional qualities of performance and usability.

We believe that a third, yet very important question should be asked: (3) how do you want to feel? The emotional factors related to software and systems are important parts of technology appropriation, as well as important in determining long-term use [24]. In the example of sharing pictures with friends, users may want to feel engaged. Offering a server where photos can be uploaded satisfies the functional and quality goals, but not the emotional goals, such as being engaged or feeling connected. For this, functionality such as tagging, commenting on, and perhaps modifying photos could be implemented.

As far as the authors are aware, there has been no research integrating emotional factors into the agent-oriented requirements engineering process. Given that many agent-oriented modelling notations capture the essence of goals, roles, qualities, and the connections between these, we choose to extend Sterling and Taveter’s notion [43] agent-oriented modelling notation to capture emotions.

### 3.1.3 Integration of ethnographic data into software engineering

Using ethnography to inform technology design is not a new concept. However, despite some excellent work in the area [15, 16, 51, 39, 42, 32], bridging the gap between ethnographers and software engineers is still in its infancy, especially for systems in the social domain [2].

Several existing methods have been successful in combining ethnography with software engineering models to produce requirements specifications. Hughes et al. [14] were among the first to recognise the value of ethnography in software design, and among the first to recognise the importance using boundary objects to allow ethnographers and software engineers to share their understanding. Viller and Sommerville [49, 50] prescribe a method for documenting the results of ethnographic studies using UML use cases and domain models. Millen [26] presents rapid ethnography, which is a collection of field work to help understand a socio-technical system for which technology is being designed. Ethnographers
observe users in the field, and answer a structured questionnaire after the observations. The resulting data is used to derive a causal model that informs the system design. Diggins and Tolmie [11] model the shared understanding of terminology between ethnographers and other field workers using *Grounded Innovation Maps*, which is a boundary object that organises ethnographic data in a manner that is palatable to both system designers and the field workers.

Many of these methods have been successfully applied to socio-technical systems, however, with a focus on functionality, they do not capture the emotional needs of users. We believe that some of the lessons learnt in our work could be integrated into these methods.

### 3.2 Motivational modelling and notation

In this section, we present the necessary background on the modelling notation that we use in our work. We believe that our ideas can be generalised to other agent-based notations.

POSE models are based on syntax and semantics of Sterling and Taveter [43]. Their aim is to make high-level models palatable to non-technical stakeholders. This is achieved by using models with a straightforward and minimal syntax and semantics, and based on the agent paradigm, which is a natural way to consider socio-technical systems. The early requirements phase models, which Sterling and Taveter call *motivation models*, are particularly lightweight, and we have adopted them due to their simple nature, and ability to easily capture the important concepts of roles, functional goals, quality goals, and the relationships between them.

**Motivational goal models.** Motivation goal models are useful in early stage requirements engineering to capture initial understandings, and share these with other stakeholders. Roles and goals capture the users and their wants, and agents (human or artificial) can then fulfil these roles. We define these concepts based on Sterling and Taveter’s definitions.

**Definition 1** (Functional goal). *Functional goals* are based on motives, and describe an intended state of the environment. Functional goals can consist of sub-goals.

**Definition 2** (Quality goal). *Quality goals* are non-functional goals (sometimes referred to as *soft goals*). Quality goals are attached to functional goals, capturing that the functional goal should be achieved while maintaining the quality.

**Definition 3** (Role). *Roles* are some capacities or positions that facilitate the achievement of functional goals. Roles are played by agents, which can be humans or artificial. Roles have *responsibilities*, which determine what the agent must do to achieve the functional goals, and *constraints*, which determine the conditions that must be considered when trying to achieve functional goals.

Figure 1 defines the notation employed by Sterling and Taveter in their motivational goal models. Functional goals are represented as parallelograms, quality goals are clouds, and roles are stick-like figures. These constructs can be connected using arcs, which indicate relationships between them. Typically, arcs are used to connect the following:

1) **Roles to functional goals**: this represents that the agent playing the role is responsible for achieving the functional goal. For example, a person playing the role of *Manager* is responsible for the functional goal of approving a piece of external correspondence.

2) **Functional goals to quality goals**: this represents that the functional goal should be achieved under consideration of this quality. For example, the functional goal should be achieved efficiently, where “efficiently” may be defined more precisely.

3) **Goals to sub-goals**: this represents that the functional goals are related. We use indirect arcs to denote that one goal is a sub-goal of the other, and therefore the sub-goal contributes to achieving the higher-level goal. To distinguish the goal from its sub-goal, we place the sub-goal below the goal, thus enforcing a hierarchical goal model, rather than adopting notation from existing software engineering notations, such as directed arcs. We have found this to be a simple and natural way of expressing sub-goals for non-technical stakeholders.

This notation is lightweight, yet flexible enough to capture the important high-level goals of stakeholders, and system qualities.

**Motivational scenarios.** Motivational scenarios also play a key role at this level. Motivational scenarios provide concrete instantiations of the system, provide some temporal alignment with the functional goals in the model, and help to capture the dynamics of the greater socio-technical system.

**Role models.** In addition, role models complement the motivational goal models and scenarios. Role models are descriptions of the roles captured on the motivational models, which include the role name, the responsibilities of the role, and the constraint under which the role may operate.

### 4 Emotion-led motivation modelling

In this section, we present our conceptualisation of emotion-led motivation models, which we call *people-oriented software engineering* (POSE) models. POSE models extend those of Sterling and Taveter [43] by adding...
adding emotional goals. We suggest additions to goal models, motivational scenarios, and role models. We believe these additions provide a flexible way to model the emotions of people and agents playing roles in a system.

4.1 Emotions in systems and software

Before we present our conceptualisation, we first define what it is that we want POSE models to capture.

Definition 4 (Emotion). An emotion is a feeling that characterises a state of mind. Examples of emotions include feeling joy, terror, safe, empowered, or normal.

We aim to produce systems that both support people’s emotional desires, and also influence their state of mind to enhance the feeling of some emotions. Our key addition to Sterling and Taveter’s models are emotional goals, conceptualised by Marshall [22]. Emotional goals are different from functional goals, which address the functional intent of users, and from quality goals, which detail the qualities of systems (e.g. “secure” or “usable”). Emotional goals address how people feel, and as such, are not properties of a system (or a goal), but of people engaged in the greater socio-technical system.

Types of emotions. We offer an important distinction between two types of emotion in the context of systems and software:

1) Personal emotions: These are the emotions that a person feels (or wants to feel) irrespective of the system being studied. Examples include emotions such as feeling loved, feeling safe, or feeling normal. Although the emotions are independent of the system, it is clearly useful to model those personal emotions that are in the scope of the system; that is, they can be influenced by or can influence the system. While a technological system can influence these emotions, they are desires of users that exist independently of any specific support that is provided.

2) Context-specific emotions: These are the emotions that a person feels (or wants to feel) about a system, or piece of technology. Examples include feeling engaged with the system, feeling frustrated by the system, or feeling that the system is integrated in their life. These are distinctly emotions about the system, and exist only in response to the system.

We believe that the above mentioned two different types of emotion are sufficient to capture the emotional desires with respect to systems, such that POSE models can inform system design.

Making a distinction between these two is important from an engineering perspective, because if some particular part of a system (or design) is removed, if it supports a personal emotional desire, additions must be made to address the emotion. However, this is not necessarily the case for context-specific emotional desires.

4.2 Emotional goals

To model emotions, we extend Sterling and Taveter’s motivational models to include emotional goals.

Definition 5 (Emotional goal). Emotional goals are non-functional goals that describe a desired reflective-level emotion of a role.

Emotional goals differ from quality goals in that they represent the reflective-level emotions in Norman’s theory [30] (discussed in Section 2), whereas quality goals (and functional goals) represent intended properties of the system that affect the behavioural levels, and possibly the visceral levels in some cases; e.g. qualities about aesthetics.

Figure 2 shows the syntax used for emotional goals, although we omit the lines whenever the relationship is clear. One can see that the relationship is ternary: emotions are attached both to a role and to a goal. In Figure 2, the emotion is related to a functional goal, but this can also be attached to quality goals and other emotional goals; e.g. a parent seeing their child excited to open a birthday gift may excite the parent themselves.

Fig. 2. Notation for modelling emotional goals

Linking of the role to the emotional goal is important, as this allows us to model conflicting emotional goals in response to the same goal for different stakeholders. That is, for a particular goal, it is natural that different stakeholders may (want to) have different emotional goals, so the emotional goals are inherently linked to the people playing the roles, not just the corresponding system property.

If the emotion is a personal goal, then the syntax from Figure 2 is read as: “Role R feels the personal emotion EG, and the goal G helps to support this emotional goal”.

If the emotion is a context-specific goal, then this is read as: “Goal G supports the context-specific emotion EG for role R.” That is, the system should influence the role R to feel the emotional goal EG via the goal G.

In our experience, we have not found it important to use notational differences between personal and context-specific emotional goals. Instead, we opt for simplicity of high-level models, rather than detail, which we find confuses stakeholders. In our projects, we have used the context in which the goal falls, or additional information in other models, to determine the type. However, we acknowledge that grouping emotional goals by type will not necessarily be sufficient in all cases.

Our proposal, shown in Figure 3, is to simply annotation context-specific goals with an icon representing a
functional goal, to indicate that the emotion is explicitly tied to the context.

As a minor addition to Sterling and Taveter’s models; our models add one new symbol, and two concepts (personal and context-specific emotions). This is certainly our aim: to offer minor variations on existing notations and methodologies in order to capture important aspects of the greater socio-technical system. We believe that the notation could be readily included into other goal-oriented notations, such as \( i^* \) [54], KAOS [9], or Jacobson’s UML use case diagrams [18].

Example — Gift giving goal model. To illustrate our notation, we adopt the gift giving model from our earlier work [27]. The model, shown in Figure 4, consists of two roles: gift giver and gift receiver. The goals of the giver are to choose an appropriate gift that makes the receiver feel special, and to present the gift in a creative way. The goal of the receiver is not just to receive the gift as one would during a commercial exchange, but to acknowledge the receipt of the gift and show that they appreciate the feelings that the giver is conveying. The emotional goals play an important part in the description of gift giving. Simply choosing or giving a gift is not the primary motivation of the gift giver. Instead, they want to choose a gift such that the receiver is made to feel special in the eyes of the giver.

Gifting is a good example to illustrate the complexities of emotions and social interaction. There are many different attributes to gift giving, and these differ over cultures and even age groups. It is not possible (or perhaps even desirable) to capture these in one single model. For example, as described by Otnes and Beltramini [31], one property of giving gifts is that the gift giver generally obtains more satisfaction from the interaction than the receiver. This is not captured by the model in Figure 4, which is just an illustration of emotional goals.

4.3 Motivational scenarios

For traceability and completeness, we also adapt motivational scenarios to consider emotional goals. Sterling and Taveter’s motivational scenarios consist of a scenario name, a scenario description, which specifies how the roles and goals come together, and a quality description, which specifies what qualities should result from these goals.

Our straightforward addition to this is to add a field to the motivational scenario tables: emotional descriptions, which describes the key emotions that are part of the scenario.

Example — Gift giving scenario. A motivational scenario for gift giving is shown in Table 2. The motivational scenario outlines the process of gift giving, and relates this to the functional, quality, and emotional goals. The key emotional properties that we aim to achieve in the gift giving motivational scenario are outlined, and these can be traced back to corresponding emotions in the goal model.

<table>
<thead>
<tr>
<th>Scenario Name</th>
<th>Gift giving</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario description</td>
<td>A gift giver chooses and presents a gift to the receiver, who acknowledges the effort that the giver put into choosing and presenting.</td>
</tr>
<tr>
<td>Quality description</td>
<td>The gift should be appropriate, and the interaction should promote a sense of sharing.</td>
</tr>
<tr>
<td>Emotional description</td>
<td>The gift giving should promote a feeling of friendship, make the receiver feel special, and make the giver feel appreciated.</td>
</tr>
</tbody>
</table>

4.4 Role models

We add emotional considerations to role models. Each role model describes the emotions that an agent playing the role may feel, and, if necessary, how the responsibilities and constraints affect this emotion. This is achieved with two fields in the role models: (1) personal emotional goals; and (2) context-specific emotional goals. These are lists describing the relationships of roles to their respective emotional goals.

In addition, the scopes of the role responsibilities and constraints are expanded to include emotional aspects. That is, a role can be responsible for performing a functional goal that affects the emotions of another role, or can be constrained by the fact that it has to consider the emotional goals of a role, including itself.

Quality goals are not part of role models because quality goals are properties of the system. However, emotions are included because they are properties related to the roles as well as the system.

Example — Gift giver role model. Table 3 shows a role model for the gift giver role. Note that the emotional goal of making the receiver feel special is considered a responsibility for the gift giver. A second role model for the gift receiver must also be considered, but it is omitted here. While this role model is straightforward and adds little to our understanding of the domain, in our case studies, role models are often used to provide more detail and more precision about the responsibilities and constraints of agents playing roles that cannot be captured in goal models.
TABLE 3  
Role model for Gift Giver

<table>
<thead>
<tr>
<th>Role Name</th>
<th>Gift Giver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>The Gift Giver chooses and gives a gift to the receiver.</td>
</tr>
<tr>
<td>Responsibilities</td>
<td>1. Choose a gift that makes the receiver feel special.</td>
</tr>
<tr>
<td></td>
<td>2. Present the gift in a creative way.</td>
</tr>
<tr>
<td>Constraints</td>
<td>The gift should be appropriate.</td>
</tr>
<tr>
<td>Personal emotional</td>
<td>The giver wants to feel close to the receiver.</td>
</tr>
<tr>
<td>goals</td>
<td></td>
</tr>
<tr>
<td>Context-specific</td>
<td>The giver should feel appreciated by the receiver’s acknowledgement.</td>
</tr>
<tr>
<td>emotional goals</td>
<td></td>
</tr>
</tbody>
</table>

4.5 Process model

In this section, we outline our process model for emotion-led requirements modelling. Figure 5 outlines the process model that we follow in our requirements elicitation exercises, which is an extension of the process outlined in our earlier work [27].

In this model, we advocate an incremental process in which different activities within the domain are studied to provide an overview of the motivations in the domain. Identifying these activities is the first step, after which we study each activity, building on the previous study. These can be studied concurrently or otherwise.

The third step is to obtain data on the activity, which is used to inform the model. In our projects, we have focused on rich ethnographic data, obtained via combinations of interviews, observations, and technology probes, however the data can be in many forms, from surveys, brainstorming, domain research, user feedback, etc.

Once data is obtained, modelling commences. Our initial focus is on the primary emotional goals and their related roles, as we believe these are the most important motivators for adopting and continuing to use systems.

Next we extract the key functional goals for the system, and the quality and context-specific emotional goals that
relate to these. In parallel, we define the motivational scenarios, which depend on all of the types of goals. Finally, the role models are completed, which involved both identifying any additional roles that are important to the system that are not identified with the personal emotional goals, and also completing the role models themselves. This process has been described in further detail in a related paper [37], in which we discuss how to increase the validity of the results using different levels of group analysis.

This process is repeated on the different activities that are identified at the start of the process. For example, if we are studying ways in which close friends interact, the activity of gift giving outlined in Figure 4 could serve as one activity, while the activity of inviting someone to lunch could serve as another. Studying several interactions of this nature provides us with a view of how friends interact, what motivates them to do so, and what their specific aims may be as part of interactions. In our earlier work [27], we discuss how studying several such activities provides us with a more valuable understanding of a domain than trying to generalise all such activities.

5 Evaluation of emotional models

In this section, we present a controlled experimental evaluation of our proposed models. The goals of this study are to answer the following two research questions:

1) Are POSE models suitable as boundary objects, compared with existing goal-oriented modelling notations such as $i^*$?

2) Do people prefer the use of emotional goals instead of quality goals for capturing emotional desires?

We compare POSE with $i^*$ because both notations aim to include the social elements of socio-technical systems, and $i^*$ is perhaps the most commonly applied social modelling language in academia. Like POSE, $i^*$ supports concepts such as functional goals, quality goals, roles, and goal decomposition, but also introduces many additional concepts, including tasks, resources, and agents, and several types of relation between these, including means-end relationships, decomposition, and dependencies.

$i^*$ models were designed as a tool for software engineers to map their ideas of how a socio-technical system operates and to define the problem to be solve. POSE models were empirically designed as a tool for software engineers and non-technical stakeholders to collaboratively define their joint understanding of a socio-technical system. Because non-technical stakeholders are generally unconcerned with the difference between roles and agents, and the different types of dependency, these more detailed concepts are not included in the POSE notation.

Our hypothesis is that POSE models are more suitable as boundary objects because the sets of concepts are constrained, allowing people to more easily engage with the models, and correctly interpret meaning, even if the notation is unfamiliar, while still maintaining high expressiveness and flexibility required for social modelling.

5.1 Experiment design

We recruited a total of 20 participants from a range of backgrounds, and administered surveys to each. Pre-experiment questions assessed their background, such as experience in a computing-related field, familiarity with the modelling notations, etc.

In our final set of participants, we had 10 participants with training or experience in computing or a cognate discipline, such as computer science, software engineering, and web design; and 10 participants without training in a cognate discipline, including occupations such law, graphic design, business analysis, and accounting. We had no participants with experience in $i^*$ or POSE, and participants were not informed as to the study goals (except for the final qualitative question).

Each participant was given a survey document that consisted of models from two domains: a meeting scheduler system, taken from Yu [52], and a patient care system, taken from Yu [53]. Each participant received one system modelled in $i^*$ and the other modelled in POSE. To mitigate potential bias, we split participants into two groups, A and B. Group A received the meeting scheduler in $i^*$ and the patient care system in POSE, while group B received the meeting scheduler in POSE and the patient care system in $i^*$. To mitigate experimenter bias, we used only $i^*$ models from existing literature, and derived models of these domains in POSE. Further, participants worked independently, and requested to only ask questions to clarify notation (although no such questions were asked). To mitigate selection bias, we recruited participants with experience in software modelling, and some without.

The survey was broken into three tasks. In task 1, participants were asked to answer five questions about the meeting scheduler system, such as which roles are responsible for certain tasks or which goals (or tasks) are required to fulfil a particular goal. In task 2, participants were asked to answer six questions about the patient care system, of a similar type to that in task 1. In task 3, participants were presented with two alternative models of a photo-sharing application: one in $i^*$ and one in POSE, each modelled by us. This was the only time that participants saw alternatives of the same domain. Participants were then asked qualitative questions about the models they had seen, and which notation they found easier to understand, how comfortable they would feel modifying such a model, and how much information each captures. In particular, the final question asked participants which type of model they thought better captures the emotional desires of people involved in systems.

Before the participants began, we gave each participant a one minute overview of the POSE modelling notation, and a two minute overview of the \(i^*\) notation. The \(i^*\) notation was afforded a longer overview because of the additional concepts it uses. In our experience and one-two minute overview is about as long as possible before stakeholders lose focus.

For tasks 1 and 2, we took two measures: time and correctness. For each question, we timed how long it took the participant to complete the task. There were no time limits on tasks. After the questions were completed, answers were judged on correctness. These two measures were proxies for how well the participants could understand the models. The questions in the tasks were such that assessing correctness was straightforward. We measured correct answers as 1, incorrect answers as 0, and partially correct answers as 0.5. A partially correct answer includes cases such as correctly listing only some of the tasks that a role is responsible for, instead of correctly listing all tasks.

### 5.2 Results

Tables 4 and 5 show the results of the experiment and qualitative questions respectively. Table 4 shows the percentage of participants who answered each question correctly, the mean and standard deviation of the number of questions correct per participant, and the mean and standard deviation required to complete the task.

From this table, one can see that on average, participants answered more questions correctly from the POSE models than the \(i^*\) models, and in a faster average time. Given that these measures are our proxies for understandability and usability, they imply that POSE models are easier to understand by participants. Over the eleven questions, eight had an equal or higher percentage of correct answers, and the overall number of correct answers given per participant was higher – in the case of the meeting schedule system, by almost one complete mark. The times were also lower for the POSE models.

We attribute these results to the simpler nature of the modelling notation, and the hierarchical layout of the models, which lead to a more natural way to read them (left to right, top to bottom). As well as the quantitative results, the participants’ comments provide further evidence of this. For example, the two quotes below are from participants who had a strong preference for POSE notation:

> Far more clear, seems to capture what is important, [and] constrains models to be sensible. [Peter, Computer science student]

> It makes implicit use of space rather than explicitly using lines. [Danny, Experienced software engineer and data modeller]

Table 5 shows the average responses for the qualitative questions as a distribution over the responses. The explanations each side of the distributions show the extreme points of the five-point scale used.

From these results, we can see several expected results. First, participants have a stronger preference for using POSE models, largely because they are clearer, the icons are easier to interpret, the models are simpler & “less busy”, and the relationships are easier to identify. We attribute this to the deliberately-constrained notation of the POSE models.

Participants were also not confident that they could modify \(i^*\) models if required, but were far more confident that they could do so with POSE models. One participant commented that they would be very confident modifying a POSE model because they were able to:

> ... understand who performs task, and whether it is essential, emotional, or desirable quality of a task. [Bree, Lawyer]

Another commented that they would be not at all confident modifying an \(i^*\) model because the model presented in the survey was:

> Extremely confusing; I have difficulty following it [Pedro, Computer Science Student].

Many participants considered that \(i^*\) models capture more information. From the qualitative comments, this is almost entirely due to the fact that \(i^*\) has more types of relationships between different elements, and little to do with additional concepts such as tasks, resources, etc.:  

> Relations and dependencies seem at least more explicit/detailed. [Phillip, Computer science postdoc]

However, several participants commented that while \(i^*\) could capture more, it may not be useful:

> Yes, but not necessarily relevant/essential information. [Tanya, Business analyst]

With regards to modelling emotional goals, there was a clear preference for the POSE models, with only one participant expressing a preference for \(i^*\)’s notation. This participant commented that they preferred \(i^*\)’s way of modelling emotion because POSE models are:

> ... organised by hierarchy, in my opinion we can’t classify emotional things and feelings like this. [Ingrid, Digital media designer]

Despite this, Ingrid preferred the hierarchical nature of POSE models.

The preference for explicit emotional goals reflects our previous experience using these models with stakeholders in projects. We have observed that people like the explicit emotional goals because emotional goals provide a clear separation from quality goals. Several participants expressed similar views:

> Love hearts indicate emotion & link back to a person represented as a figure. [Bree, Lawyer]

> Easy to misinterpret the \(i^*\) emotions as other soft goals. [Nathan, Computer Science postdoc]
### TABLE 4
Task analysis results

<table>
<thead>
<tr>
<th>Notation</th>
<th>Percentage of correct answers</th>
<th>Mean Score</th>
<th>Stdev Score</th>
<th>Mean Time</th>
<th>Stdev Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Q1</td>
<td>Q2</td>
<td>Q3</td>
<td>Q4</td>
<td>Q5</td>
</tr>
<tr>
<td>Meeting scheduler</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i*</td>
<td>40.0%</td>
<td>25.0%</td>
<td>55.0%</td>
<td>60.0%</td>
<td>70.0%</td>
</tr>
<tr>
<td>POSE</td>
<td>70.0%</td>
<td>60.0%</td>
<td>80.0%</td>
<td>75.0%</td>
<td>60.0%</td>
</tr>
<tr>
<td>Patient care</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i*</td>
<td>70.0%</td>
<td>60.0%</td>
<td>80.0%</td>
<td>50.0%</td>
<td>50.0%</td>
</tr>
<tr>
<td>POSE</td>
<td>60.0%</td>
<td>80.0%</td>
<td>70.0%</td>
<td>80.0%</td>
<td>75.0%</td>
</tr>
</tbody>
</table>

### TABLE 5
Qualitative questions

<table>
<thead>
<tr>
<th>Property</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>++</td>
</tr>
<tr>
<td>Understandability</td>
<td>i* preferred</td>
</tr>
<tr>
<td>Modifying i*</td>
<td>Not at all confident</td>
</tr>
<tr>
<td>Modifying POSE</td>
<td>Not at all confident</td>
</tr>
<tr>
<td>Concepts</td>
<td>i* captures more</td>
</tr>
<tr>
<td>Structure</td>
<td>i* preferred</td>
</tr>
<tr>
<td>Emotions</td>
<td>i* preferred</td>
</tr>
</tbody>
</table>

Interestingly, there is no significant difference in the results between those participants with training in computer science or software engineering, and those without. That is, participants familiar with modelling notations such as UML, ER modelling, etc., still preferred the simplified view.

### 5.3 Threats to validity

There are two main threats to external validity. First, only two domains were modelled and used as part of the experiment. Additional studies would improve the generalisability of the results. Second, while 20 is a reasonable number, the sample is biased towards professionals, mostly with university degrees in areas where modelling notations such as decision trees are used, and did not include other representatives, such as children or older people.

A threat to internal validity is that time and correctness were used as proxies for understandability and usability, and may not be entirely accurate. Another internal threat is the questions posed, which may have potentially biased the results towards POSE. However, we explicitly included questions that we believed would be easier to answer in i*, such as identifying which things a role was responsible for (Question 5 of the patient care system), which we believed would be straightforward to identify in i* due to their use of actor boundaries.

### 5.4 Discussion

The results of this evaluation support our hypothesis that POSE models are suitable boundary objects. With one of our primary criteria being a modelling notation that is palatable to non-technical stakeholders, many of whom will not have the desire to learn a new notation, it is paramount that a simple and clear notation is used. On top of this, we did not see a large divergence in results between participants with computer science training and those without. That is, the participants with a computer science background also preferred the POSE notation over i*. From this, we conclude that POSE models have the capability to carry the voice of the user, and keep many stakeholder groups involved throughout the requirements process.

### 6 Case study — Emergency alarm systems

In Section 5, we demonstrated the general suitability of POSE models for emotional modelling. In this section, we extend this work to a case-study evaluation of POSE models on a real development project. The case study is of emergency systems, discussed in the introduction. The primary focus of this evaluation is to answer the following qualitative research questions:

1) Do our motivation models capture the important emotional aspects of peoples’ needs?
2) Does the consideration of emotions provide us with useful information to improve systems and software?
3) Are the resulting models useful as communication tools between stakeholders, including software engineers?

Thus, this evaluation is a preliminary investigation into the use of emotional goals, and emotions in general, for improving software and systems.

### 6.1 Emergency systems

Our case study is that of emergency systems. The aim of these systems is to support a person, generally an older person, to remain living at home longer.
Emergency systems typically have two features: (1) an emergency alarm: the older person can raise an alarm if they require emergency attention, via e.g. pushing a button on a pendant worn around their neck or wrist; and (2) a wellbeing check: the older person informs a service provider that they are well, on a daily basis, via e.g. a button on a base station. If no indication of wellbeing is received during a specified period, the service provider initiates checks on the older person.

The emergency system is installed in an older person’s home, and data from the system is monitored by a service provider. If the older person raises an alarm, the service provider contacts the person to ask if they require help, or to check that the call is not a false alarm (accidental triggering of the alarm). If they fail to make contact, or they make contact but the older person is in need of help, the service provider calls a nominated relative or friend to provide assistance, and ultimately, emergency services (e.g. ambulance) if none of the nominated contacts can attend.

For the wellbeing check, the older person is required to register their wellbeing each day, generally between a fixed period of time; e.g. 7am–11am. If they do not check in by the end of the period, the service provider calls the older person. In most cases, the older person has simply forgotten to press the button. If they cannot be reached, a similar process as for raising an emergency is initiated.

6.2 Method

To uncover user needs and emotions in the emergency system domain, we undertook an ethnographic study using range of data collection methods, primarily interviews and participant observation. We interviewed a range of people about emergency alarms and technology in general, we modelled the findings using POSE models, and used these models as communication tools with various stakeholders. This section details the study design.

6.2.1 Study design

To answer the research questions identified at the start of this section, we designed the study along the following steps:

1) We modelled our initial thoughts on the relevant functional, quality, and emotional goals for emergency systems, based on existing literature and any anecdotal stories that we had heard. This gave us a baseline model against which to compare our final model, and helped to guide the interviews (research question 1).

2) We interviewed 12 participants, categorised into three groups: (a) older people who have or had previously had an emergency alarm (four people); (b) relatives of older people who have or had previously had an emergency alarm (four people); and (c) older people who had never had an emergency alarm (four people).

3) The interview data was analysed using ethnographic content analysis to extract the key themes or aspects, such as functionality, qualities, and emotions desired by different stakeholders.

4) The key themes or aspects were modelled using POSE models (research question 1).

5) The emotional goals were used to inform new functional and quality goals for the model, and to then provide some high-level design concepts for new emergency systems (research question 2).

6) Finally, the models were used as communication tools between the research team, our partners, and a separate team of software engineers who developed a new prototype for us (research questions 2 and 3).

6.2.2 Participants

Our primary data collection technique was face-to-face interviews. We interviewed 12 people about their use of technology in general, and on emergency systems. All participants lived in Melbourne, Australia. In addition, we took notes on the interview participants and their domestic environment, such as where their emergency alarm was located.

As mentioned previously, our interview participants consist of three broad groups:

1) Four participants were older people who either have or have had an emergency system installed in their home. Their ages ranged from 85–91. Three of the four participants lived on their own, while the fourth lived with their spouse. All were supported in their wellbeing by family, friends, and neighbours, and all had the emergency system as additional support.

2) Four participants were family members (either children, nieces, or nephews) of older people who either had or have had an emergency system installed at home. In all cases, the older person lived on their own during the period in which they had the system installed, and all had the emergency system as additional support.

3) Four participants were older people who never had an emergency system installed in their home, and had not used one before. Their ages ranged from 69–79. All participants in this category lived with their spouse. From this group, we expected insights from people who are not taking up emergency systems.

6.2.3 Data collection

The main source of data collection was via semi-structured interviews with each of the participants. We interviewed each participant once, with each interview taking between 30–60 minutes. The interviews were focused on the three different types of goals: functional, quality, and emotional; and how these should be played out in emergency systems and in technology in general. Some participants were contacted for follow-up questions. All interviews were recorded and transcribed.
Interviews were semi-structured, allowing the participants to explore their thoughts and feelings. Each interview followed a similar pattern of asking three broad types of questions about technology in general, and about emergency systems:

1) What should technology (emergency systems) do for you? This question relates to functional goals.
2) How should it be? This question relates to quality goals.
3) How do you want to feel? The question relates to emotional goals.

These questions were not asked directly as they are stated above, but were based on these themes.

For those participants with experience in using emergency systems, we also asked questions regarding the contexts surrounding the emergency system, including why they signed up, what problems they experienced using the system, the key benefits of the system, what actions they believe are required to use the system, and what they believe happens when they initiate an emergency.

6.2.4 Data analysis

The interview data was analysed using ethnographic content analysis according to Patton [34]. This involves careful reading of the interview data, highlighting important and recurring themes from the participants’ responses. It identifies core consistencies and meaning of the data. At least two researchers would analyse each interview, and would compare their findings to reach an agreement on the important themes. This summary data was then presented to the research group, consisting of six people, and was subject to further analysis in the group until we reached agreement on the important themes over all interviews. The focus of this analysis was on the emotional and quality attributes of the emergency systems.

Pedell et al. [37] provide further details of the field study, including the ethnographic method followed, information about the participants, the questions, and the findings, and also how the ethnographic data informed and supported the models.

6.2.5 Limitations

There are several limitations to our evaluation. First, we consider only one system, and as such, generalising to all systems is not possible. Second, we have not repeated our evaluation, so cannot determine how repeatable it is from a software engineering perspective; that is, whether another team would come up with the same models from the same data. Despite this, content analysis [34] is a well-accepted ethnographic technique, and informs much of this part of the approach. Finally, the case study is subject to bias, in that we knew that our initial baseline models would be used for comparison, which may have distorted our approach.

7 RESULTS AND LESSONS LEARNT

In this section, we present the results of the emergency system case study, and discuss some of the lessons learnt. We divide the presentation into what we learnt about emergency systems design, and what we learnt about our modelling approach.

We briefly discuss the design of a new emergency system based on our findings, which we have prototyped and evaluated. Pedell et al. [37] present more detailed results and discussion about the design of emergency systems, the prototype, and the resulting prototype evaluation. Our focus in this paper is on the emotional goals and related models.

7.1 Emergency systems

First, we discuss what we learnt about the design of emergency systems. We present this by contrasting the models that we derived prior to the interviews with the models that were informed by the interviews. These goal models provide a sufficient overview of our particular findings. Figure 6 shows the goal model prior to the interviews, while Figure 7 shows it after the interviews. We present our findings based on the changes and otherwise between these two models. While functional and quality goals have also changed, our analysis here is limited to emotional goals. In Figure 7, we highlight changes made to the emotional, quality and functional goals.

In this paper, we concern ourselves only with the emotional goals. For each finding, we present a short quote from interviews with participants, to supports this finding, however each emotional goal is backed by additional ethnographic data. Further details are available in Pedell et al. [37].

7.1.1 Confirmation of existing emotional goals

Reassured. Our first model (Figure 6) seems to correctly capture the idea that the primary motivation for wanting to install an emergency system into a relative’s home is to provide the carer or relative with reassurance that the older person is safe. The interview data confirms this. For example:

We just felt that if she had a fall or something happened to her ... we would at least know that she was OK [Christine – daughter of an emergency system user].

Safe. Feeling safe is a key theme for the people using the pendants:

... but basically because I’ve got this, I’ve got confidence when I’m on my own [Jane – emergency system user].

Independent. From anecdotal evidence, we had learnt that independence was a key theme in emergency systems. However, we learnt the perceptions of independence vary greatly between different people who had an emergency system.
On the one hand, emergency systems were seen to preserve independence for many people who would otherwise have to live in a shared home, or with relatives:

*I think she felt that she’d got her independence back. “I can stay here by myself because I’ve actually now got something that if something happens to me, she is not going to fret”* [Feelings of her mother described by Christine].

On the other hand, emergency systems were seen by some as a threat to their independence:

*It threatened her independence. She felt that it branded her in a way that made her less independent* [Feelings of his aunt described by Joe].

Thus, some people felt that having an emergency system attracted a stigma that they could not take care of themselves because they were old.

We identified a clear distinction between these two groups. The group that felt the emergency systems gave them a feeling of independence had asked to have it installed. That is, they felt that they could no longer live independently without support for their wellbeing, and this system gave them their independence by allowing them to stay in their home for longer. The group that felt their independence threatened had been told by their relatives that they would need one. That is, the relative gave them an ultimatum of living in a nursing home, or getting an emergency system.

Those people who felt the system threatened their independence tended not to wear the emergency pendant. Joe, who insisted that his aunt have an emergency system installed, described that the two times that his aunt fell over and was unable to get up — in one case, breaking her hip — she was not wearing her pendant either time, and had to wait until neighbours or Joe himself visited. His aunt later admitted that she only wore the emergency pendant when she knew Joe was coming to visit.

### 7.1.2 Removal of existing emotional goals

**Accessible, unified, and natural.** The three context-specific emotional goals in Figure 6 that we had initially considered as important did not reveal themselves as themes from the interview data. We note that these were the only context-specific emotional goals that we had considered, and thus, these were all different to those of our participants. This results demonstrates the importance of stakeholder input for these types of systems.

### 7.1.3 Addition of new emotional goals

**In touch — personal goal.** A new theme was the feeling, for the older person, of being in touch with others — especially family and friends, which is modelled in
Figure 7. We expect that this is especially prevalent in our interview data because most of the emergency system users in our study were living alone, and felt isolated — something that is common with many emergency system users in Australia. This more general emotional goal of being in touch is supported by the emergency systems in a small way, in that the older persons felt that they were not totally alone:

Very comforting that you’ve got that contact, otherwise you’re on your own [Jane, emergency system user].

Two participants noted that they enjoyed talking with the support staff, even if over the phone, noting how courteous and caring they were.

**In control — context-specific goal.** One theme that was identified was that of being in control. Participants stressed the importance of being in control of their lives (not just in control of technology):

Technology ... should enable me to do things that I can’t do otherwise. It should not be an instrument of control or surveillance [Martin, non user].

The emergency systems were seen to be a source of resentment regarding the control in their life, which stressed out some users:

We actually turned off the services at the end of the month, because we were not happy with it. It was stressing my mother, it was stressing me [Lucas, son of emergency system user].

The feeling of being in control is closely related to that of feeling independent. Our analysis showed that some people could feel independent, but not have the feeling of control over the emergency system. For this reason, we modelled the feeling of “in control” as a context-specific emotional goal (Figure 7) — while an older person wants to feel in control of their own life, an emergency system cannot give them this control, but should be designed to give them control of the system itself.

**Integrated — context-specific goal.** A recurring theme about emergency systems and technology in general is that they should feel integrated into one’s life, modelled in Figure 7. For example, in the context of emergency systems, any disruption that it would cause to the older person’s regularly daily routine was considered a negative:

The biggest problem is it didn’t get integrated into my mother’s life and part of the routines that she had. If I use jargon, she didn’t ever really appropriate the technology. It didn’t become part of her life. And she never really viewed it positively [Lucas, son of emergency system user].

### 7.1.4 Modification of existing emotional goals

**Cared for → Cared about.** An important finding of our study is the feeling of caring. We initially modelled feeling cared for as an emotional goal in Figure 6. However, from our interview data, it became clear that none of the
older persons wanted to be cared for. In particular, two older women who had worked in the aged care sector all their career did not want to receive care:

*She worked for disability services all her life. This transition was difficult for her, because she went from being the carer to being someone who didn’t want to be cared for* [Joe, nephew of emergency system user].

Instead, the participants in our study were more concerned with being cared about, which has been modelled in Figure 7. The older persons appreciated that their families, friends, and neighbours cared enough about them to want to them to be safe:

*We were actually clear on the objectives of why we were doing it. And, again, that was an easy message, saying, “Look, we’re not here to alarm, but we’re doing it because we care”* [Lucas, son of emergency system user].

This theme was common, and some participants felt that older people were treated patronisingly when getting care they did not particularly want. The distinction is important for the interaction design of systems: the systems should support the notion that people care about the older person, while not making them feel like patients.

### 7.1.5 Summary of findings about emergency systems

From the interviews and analysis, we learnt several important considerations about emergency systems, and technology design in general. As well as the obvious feeling of safety, two prominent emotions were that the older person wanted to feel in control of the system, and more generally, they wanted to feel in touch — something that emergency systems could contribute to, but do not currently address particularly well. In addition, we learnt that the older people did not want to be cared for, but they wanted to be cared about.

### 7.2 Design improvement

In this section, we discuss the design of a prototype emergency system that we believe fulfils some of the key emotional goals. A more detailed description of this design and how it came about, as well as detailed results of a field study trailing this prototype, are presented in other work [37].

This prototype is not intended to offer a serious and complete replacement for emergency systems, but instead demonstrates how emotional goals, and the ethnographic data supporting them, can be used to guide the design process, and how more general socially-oriented systems can achieve some of the goals of emergency systems.

This improvement focuses on adding new functional and quality goals to complement or fulfil some of the emotional goals identified. Our particular focus is on the emotional goals of the older person, as the emotional goal of feeling reassured for family members and carers is strongly supported by existing emergency systems, so the design is based around the wellbeing check.

#### 7.2.1 System design and implementation: staying in touch

We have implemented a complete prototype, with a particular focus on the *I’m in touch* goal from Figure 7. The prototype contains (among other functionality) a digital photo frame on a tablet, which displays photos sent to it from relatives and friends, via email. The older person can place the frame in a location of their choosing, and can view photos, comment on photos, which sends the comments back to the sender (these can be further replied to by the sender), and scroll through older photos. When the older person has not interacted with the frame for a specific amount of time, where the time is configurable for each person, the carer can contact the person. The older person can press a button that is the same as the previous wellbeing check. Additionally, the tablet’s accelerometer is used to detect movement, which is logged as interaction with the system.

Any interaction from the older person establishes that they are well, so the wellbeing check is not required if they move or interact with the picture frame. Settings can be changed, so that instead of the older person being required to interact within a fixed period, they can specify different time periods, giving them control.

Allowing the sender of the message to acknowledge a response via another message permits a two-way communication, which fulfils the emotional goals of feeling in touch and cared about (Figure 7).

As an illustration of how we use POSE models to design more specific requirements for systems, Figure 8 shows part of a goal model for our new system. This goal model breaks down the goal *I’m in touch* into more specific goals that are closer to functional requirements. This model is not complete, but is for illustration purposes only.

We have translated the goal of feeling “in touch” directly into a functional goal called *I’m in touch* (Figure 7). The primary idea is that instead of sending a one-way signal via a button, the older person using the emergency system to stay in touch with their family, who can respond (and therefore acknowledge) this, delivering a sense of being in touch, while also fulfilling the goal of communicating that they are well.

In Figure 8, we decompose this functional goal into a more refined set of goals that are specific to our product design. For example, to communicate with a carer or relative, an older person can look at a picture (implicitly communicate), or they can like the picture or send a message (explicitly communicate).

The design was inspired by our participants, many of whom talked about how much their family means to them, and proudly displayed photos of their families on tables, bookshelves, etc. In particular, one participant hid
his emergency system base station by covering it with photo frames containing pictures of his grandchildren.

7.2.2 System evaluation

This prototype was placed into the homes of nine older people over a period of approximately two weeks each. Interviews were conducted both before and after the deployment period to evaluate how well the design and implementation fulfilled the goal model from Figure 7. The evaluation demonstrated four key findings:

1) Those older people who had or have had an emergency alarm reported a more positive experience using the prototype developed according to emotional goals than using their existing alarm.
2) The older people enjoyed the social interaction of viewing photos and sending messages to their carers.
3) The benefits of the emotion-led design are particularly apparent when the users are not particularly tech-savvy, or when they are an existing emergency system user.
4) Some participants still did not feel in control, as they did not want carers to be informed about their actions.

Further details on the prototype, on the evaluation study, including the results are available in Pedell et al. [37].

7.3 Emotion-led modelling

In this section, we discuss some of the lessons that we learnt about our emotion-led modelling.

7.3.1 Personal vs. context-specific emotional goals

In Section 4.2, we made a clear distinction between personal and context-specific emotional goals. In previous work on emotions and emotional goals [3, 5, 40, 47, 21], we have not seen such a distinction made.

It was not until we attempted to model the outcomes of initial brainstorming session for the emergency system case study that this difference was identified. When analysing and finding the key emotional factors, it became clear that some factors were personal and persisted outside of the scope of emergency systems, and some were tied directly to the system itself.

In our initial attempts to model these two distinctions, we used notational differences, such as placing the role between the functional goal and emotional goal for context-specific emotional goals. However, as we progressed further, we realised that our research group and other stakeholders could determine the difference from the context, so we used one syntax for both.

7.3.2 All quality goals are implicitly context-specific emotional goals

In both our initial brainstorming and our data analysis, we found that many of the quality goals that were derived also came up as emotional goals — mainly context-specific emotional goals. For example, the goal of the call for help being immediate was derived as both a quality goal and an emotional goal from different analyses. This is consistent with Norman’s theory [30]: events are processed at one level, but the emotional response can trigger responses at other levels. Therefore, if one desires a particular quality in a system, then
they also want to feel (or perceive) that the system has that quality. In short, each quality goal has an implicit emotional goal associated with it.

In initial data collection, we often categorised both, however, during modelling, we categorise these as either quality goals (related to behavioural level) or emotional (related to reflective). In the example of an immediate call for help, this is clearly related to system performance, so is behavioural. However, there could also be a clear reflective emotional response to this, and quality goals in models can be read this way.

7.3.3 Emotion-led models as analysis tools
The resulting model in Figure 7 is simple, and conveys only part of the important information of the domain; e.g. we supplement the goals with quotes in Section 7.1 to provide additional context. While it is easy to dismiss the model as being superfluous in the context of surrounding discussion and quotes, this does not consider that the quotations and analysis of them are a result of the modelling. That is, we use a grounded analysis to sift, select, and organise the many pieces of information and abstract them into a sensible and coherent grounded theory, which we record using our models. The quotes in the paper were selected because they represented particular emotional goals, rather than being picked directly out of the large amount ethnographic data.

Our transcribed interviews consisted of almost 40,000 words. We found that the modelling provided a useful, structured way to sift and organise ethnographic data. The difficulty is, of course, in gauging the right level of detail, and deciding what should be discarded, what should be included, and what should be merged with other concepts. Including all relevant detail would result in a unwieldy model that is not useful for communication, while including too little would result in a meaningless model. Having validated the emergency alarm models with several participants, we believe that our models are at an appropriate level for understanding the problem domain.

7.3.4 Emotion-led models as communication tools
In much of our previous work [28, 27, 36], our focus has been on the use of agent-oriented models as shared communication tools between stakeholders. In our experience, the lightweight notation and focus on high-level concepts make them useful in early-phase requirements engineering.

This section presents our experiences in using this novel emotion-led models for communicating with stakeholders in this project.

Communication within the research team. The models were used extensively and effectively within the research team to keep track of our current understanding of the domain. As well as providing a way to document our understanding, they were used in meetings to discuss findings, design ideas, and future work.

Communication of results. Our POSE models were used as a way to communicate our results, much as they have been in this paper. As well as a presentation of the model to record our understanding, we presented reports to one of our funding bodies based on the structure of the model. That is, headings in the report were taken directly from the model, and discussions centred around the goals on the model. In addition to providing a structure for reports, we found that this provided readers with a clear summary of results that could be referred to in order to put the results in perspective.

Communication with developers. Our POSE models were given to the team that developed our prototype. As well as giving the development team an understanding of the domain, the models helped us to provide them with clear motivation for the functionality that they were implementing.

One unexpected use of the models in this context was the developers employing them as a project management tool. The development team, who are separate from our research team, use a lean software development process, and, roughly speaking, manage progress by noting which stage of completion project tasks are in. The development team used the goal model to derive the tasks, adding each goal as a task to their project plan, and “ticking off” goals once the corresponding task is complete.

Although we have used the goal models for similar purposes in the past, and Marshall [21] reports on using goal models for project management, the development team on the project were unaware of this work. Clearly, the goal models worked for them, and were used directly in the development.

Communication with aged-care providers. Although we have not typically shown the models to aged-care providers, in discussing our work with them, the models and the particular abstraction we have chosen continues to provide a structured way to communicate what we have learnt from the project.

Abstraction and hierarchy. Consistent with our previous work [28], the higher-level graphical goal models were more useful as shared artifacts in meetings and discussions than the more detailed role models. Further, the hierarchical nature of the goal models allowed us to focus on the highest-level models for the most part, and then drill down into more detail at the lower level for more specific discussions.

Notation. The emotional goal notation is straightforward. However, one problem that we are encountering in our projects is that they result in a significant increase in repeated role figures and arcs on the models, compared to models without emotional goals. That is, they do not increase the number of roles, but result in repeated roles on models, or additional arcs that cross each other. We identify three reasons for this: (1) we often represent additional information that we did not otherwise consider, by asking additional questions about emotions; (2) the notation of an emotional goal requires three graph
nodes (a role, an emotion, and a functional goal), and two arcs; and (3) it is often the case that emotional goals are attached to functional goals in which the roles that achieve the goals are not the same as the role that has the emotional goal. For example, in the gifting model in Figure 4, the roles attached to the emotional goals are not the same as the roles that achieve the functional goal that fulfils those emotional goals.

Despite this, our models have not become overcomplicated in our view, because we use the hierarchical nature of the models to split them into several smaller models. We also avoid excessive arcs by implicitly using space to convey the relations instead of lines. In our experience, this is possible in most cases, and our experiments from Section 5 demonstrate that it is preferred by many participants.

**Summary of communication benefits.** Overall, we found our goal models to be effective communication tools between different stakeholders. The models do not capture all of the ethnographic data, nor all details required to build a system. However, as tool for representing a shared understanding of the problem, we found them useful.

We ultimately did not show the goal models to our participants. However, we are currently working on other projects in which end users are encouraged to create and edit goal models. We are finding that models that include emotional goals resonate with stakeholders — especially non-technical stakeholders. We are exploring areas such as depression care, sleep disorders, and other areas related to wellbeing, and are finding that non-technical stakeholders embrace POSE models. The idea of considering emotions at an early stage resonates especially strongly in these areas in which the user’s state of mind is paramount.

**7.4 Discussion**

In this section, we reflect on the research questions identified at the start of Section 6.

**Can our emotion-led goal models capture important emotional aspects of peoples’ needs?** Based on our findings, we captured many of the important emotional aspects for the emergency systems. As well as extracting the important themes from the ethnographic data, the evaluation of our prototype showed high user satisfaction, especially for the group of users who had previous experience with an emergency system.

Our models do not capture all relevant information, and further, the quality of the models is only as good as the data that backs them. However, we believe that the high-level view of motivations, including functional, quality, and emotional goals, is an important factor in software and systems engineering.

**Does the consideration of emotions provide us with useful information to improve systems and software?** Evaluations of our prototype provide support that that the consideration of emotions can lead to improved systems and software, and ultimately, better outcomes for users. Further evidence for this is the continued engagement from our industry partner who develops software for emergency systems, and interest from several new partners in the aged-care service industry.

While we believe that considering emotions would have the most visible impact in socially-oriented systems, we also believe that the consideration on emotions can have an impact on systems employed in organisations, which is consistent with related findings [25].

**Are the resulting models useful as communication tools between stakeholders, including software engineers?** While we can offer only anecdotal evidence of support for this question, our experience in this and other projects indicates that the most valuable aspect of POSE models is not as a summary of the ethnographic field data for our own purposes, but as a tool to share our understanding of this data with others. Being able to present a 1–2 page diagram outlining the most important motivations of users is key to communicating and motivating system design. The use of the models by the team that developed the prototype further demonstrates the value of the models for organising and understanding the key motivations.

**8 Conclusions**

In this paper, we introduced a straightforward notation of modelling emotional goals in agent-oriented modelling. We distinguish between two types of emotional goals: personal emotional goals, which model the emotional desires of users; and context-specific emotional goals, which model the desired effects that a system has on its users. These models were evaluated both via a controlled user study, and in the context of emergency systems, demonstrating that emotional goals can capture the key emotional desires of users, and can lead to improved design of systems and software. We have designed, implemented, and evaluated a new prototype for an emergency system, focusing on the “wellbeing check”, and demonstrated that asking a few simple questions regarding users’ emotions can lead to improved outcomes of systems.

Based on our project, we have attracted further funding from grant schemes and industry partners to apply our emotion-led models in areas such as depression care, support for sleep deprivation, and home care for the elderly, and applications for promoting low-carbon living. While some of these are currently undergoing trials that appear to be successful, further work needs to be done.

Our future work will focus on mapping the idea of emotional goals from requirements to software design, implementation, testing, and system evaluation, allowing us to trace emotion-led requirements throughout the software development lifecycle. Furthermore, we will aim at gaining deeper understanding on how we can map emotional goals to system designs, providing more systematic methods for considering emotions in design. Finally, as a method of validation and repeatability of our
modelling approach, the study data will be interpreted and modelled by others.

In recent work [19], we have considered how to accommodate the differing reactions between individuals at the reflective level. We used the notions of personas [8] and scenarios to evaluate how different people may react to the same designs. Cooper [8] defines personas as hypothetical archetypes for actual users. For several projects, we have used ethnographic data to derive several rich, realistic personas representing different people who hold a stake in proposed systems. We then walk through pre-defined scenarios to help us reason empathetically in place of those personas, improving our evaluation of design decisions, and potentially leading to new design ideas.

Acknowledgements

This research is funded by the Australian Research Council Discovery Grant DP130102660 and the Smart Services CRC.

REFERENCES

gating the influence of personal values on requirements for health care information systems, 2011.


Author/s: Miller, T; Pedell, S; Lopez-Lorca, AA; Mendoza, A; Sterling, L; Keirnan, A

Title: Emotion-led modelling for people-oriented requirements engineering: The case study of emergency systems

Date: 2015-07-01

Citation: Miller, T; Pedell, S; Lopez-Lorca, AA; Mendoza, A; Sterling, L; Keirnan, A, Emotion-led modelling for people-oriented requirements engineering: The case study of emergency systems, JOURNAL OF SYSTEMS AND SOFTWARE, 2015, 105 pp. 54 - 71

Persistent Link: http://hdl.handle.net/11343/54989