Modelling Emotional Requirements

Antonio A. Lopez-Lorca*, Tim Miller†, Sonja Pedell*, Leon Sterling* and Maheswaree Kissoon Curumsing*
* Swinburne University of Technology, Hawthorn, 3122, Australia
{alopezlorca,spedell,lsterling,mkissooncurumsing}@swin.edu.au
† University of Melbourne, Parkville, 3010, Australia
tmiller@unimelb.edu.au

Abstract—The way people feel about a technology can determine whether the technology is embraced or rejected by its intended users. People’s feelings and emotions are particularly important for uptake of socio-technical systems involving social behaviour. For example, a photo sharing web application will not be used if people do not feel engaged or in-touch with their friends and family during use. Considering the feelings of the intended users of a system can uncover new requirements, leading to an improved and more accepted system. We contend that requirements engineering, in developing systems for the new social age, needs modified lightweight practices. In this paper, we propose adapting a requirements engineering methodology to include emotion modelling for socio-technical systems. We extend a set of existing agent oriented lightweight models to consider emotions, describe the process for modelling, and demonstrate both models and process in an example of personal alarm systems for older adults.

Index Terms—Emotion modelling, requirements engineering, agent oriented modelling, emergency alarm systems

I. INTRODUCTION

Software engineering methodology has traditionally focused on the functional and non-functional requirements of systems. The emotional desires of users (i.e. the feelings that characterise their state of mind) in relation to the use of the system are often ignored. Unfortunately, a system can be developed that, although compliant with the requirements, does not fulfil the needs of their users.

We suggest that one way to produce better systems is to explore the users’ emotions to uncover those hidden requirements that may otherwise be ignored. While there have been approaches that model user perspective under the general label of soft-goals [13], existing work including our own [2, 3, 8, 11], indicate that emotions need to be treated differently from traditional quality goals, because they are properties of the users, not of the system.

Consider the example of photo-sharing web applications, such as Instagram and Facebook. If a team of software engineers is tasked with the goal of developing a system for sharing photos, simple systems such as FTP or email would fulfil the functional goals such as uploading photos, sharing with specific individuals, and commenting on photos, and would fulfil quality goals such as sharing efficiently and securely. However, such a system would be a complete failure, as it would fail to address the social requirements, such as being engaging or making the user feel socially-connected with their friends. We treat such requirements as emotional goals, from the fact that they inherently include the user, rather than just the system itself.

In previous work [3], we proposed the use of emotional goals to capture and model the most important emotional aspects of socio-technical systems. We represented people’s emotions explicitly separated from other type of goals (functional and quality), finding that this lead to new requirements and an improved user experience in the domain of personal emergency alarms. As a personal choice and a matter of style, we adopted and extended the agent-oriented modelling notation of Sterling and Taveter [9]. These models provide a lightweight representation of the system to support the discussions with stakeholders and to better understand the system to be developed. Our previous experience demonstrates that the number of errors introduced during the analysis stages of development can be reduced by engaging stakeholders for as long as possible [4]. We believe that using agent-oriented modelling is suitable for modelling the social aspects of socio-technical systems, as they use concepts such as roles and goals, which are a natural abstraction of the real interactions between people and systems.

In this paper, we extend our earlier work to consider emotions as part of a complete requirements engineering methodology. We apply these models on the case study of a personal alarm used by older adults that live at home independently.

The rest of the paper is structured as follows. In Section II, we discuss existing work relevant to our approach. Section III introduces our extended models. Section IV describes the methodological process for our modelling. In Section V, we present an application of our process to a case study, and discuss our experience. In Section VI, we discuss our approach, and then conclude the paper in Section VII.

II. RELATED WORK

There are few studies in the literature that specifically address the relation between emotions and software development. Thew and Sutcliffe [11] identify soft issues — stakeholder’s values and motivations — that have implications for the requirements engineering process. They propose a method to distinguish between soft issues that are critical for the success of the project and others that have lower impact. The types of issues that they identify include the fear caused by the possibility of project failure, and the frustration of the client because they do feel their values have been ignored. While
we acknowledge the value of considering these factors, we believe that the impact of emotions extends far beyond the development into the use and appropriation of the system. This implies that requirements engineers need to ask different questions that focus on user emotion, not just functionality.

Ramos et al. [8] argue that emotions impact user experience, as much as functionality, performance or user interface aspects. They present examples from literature and their own experience in which emotions had a definite effect on the software being developed. These examples include an error logging functionality that had to be removed from a project because the users were stressed of being traced as the origin of the mistakes, or an enterprise resource planning system developed in house in favour of an off-the-shelf system, thus allowing the developers to feel indispensable. Ramos et al. advocate integrating a trained psychologist in the requirements elicitation team to help to elicit emotional requirements.

Proynova et al. [7] argue that in health-related systems, the main benefit is social, not of economic nature — i.e. the patient recovers. Therefore, requirements engineering should be approached differently than in business information systems. In fact, current healthcare applications that simply adopt a business-like development approach have low acceptance amongst clinicians [12]. Proynova et al. use the stakeholders’ personal values to discover requirements that could remain hidden, and define features that are mapped to prefixed personal values, such as achievement, benevolence and tradition. A requirements engineer then translates these features into requirements within the context of the project. We build on the work of Ramos et al. and Proynova et al. by demonstrating how to capture and model emotional requirements in socio-technical systems.

These approaches focus on the traditional textual representation of requirements. However, an alternative more aligned with the philosophy of agile development is the use of graphic models. In our previous work, we used quality goals to capture social properties of socio-technical systems, including notions such as having fun or playing, which are more related to users than to the system itself. Similarly, Yu [13] uses soft-goals to justify why some decisions are made, based on the idea of satisficing (portmanteau of satisfy and suffice) or good enough. Soft-goals seem to be treated here as a miscellaneous category, as they include quick, low effort, user friendly, convenient and assured of attendance of important participants to the meeting. While some of these could be categorised as quality requirements as per ISO/IEC 9126, for example quick, low effort and user friendly, others such as assured of attendance fall within the emotional realm and are harder to satisfy. However, based on extensive work on quality requirements, we agree with Marshall [2] that it is important to represent emotions separately from quality goals because emotional goals are properties of the user, not of the system [3].

III. Modelling Emotions

The models defined by Sterling and Taveter [9] and refined in this section circumscribe to the early stages of the software development lifecycle. They are particularly useful to motivate the system, understand it and initiate its design. In this paper, we do not cover the transition of the modelled emotions to the detailed software design (see a discussion regarding our future work in Section VII). In Section IV we will elaborate the modelling process.

Sterling and Taveter propose lightweight models that introduce a minimum overhead to the development process while contributing with significant benefits to the resulting system. The model descriptions are guidelines written in the spirit of being as helpful as possible. We believe that these models can, and should, be adapted to accommodate existing practices and development styles of practitioners that decide to use them, and that some models will be more useful than others in different projects. We encourage adopters to use only those models that best fit their needs.

A. Emotional goals

Due to space restrictions, this section will only discuss the modifications made to the models proposed by Sterling and Taveter [9]. As we do not change substantially the character of these models, we refer the reader to this reference for further detail. The most significant aspect that we add to the original set of models is the concept of emotional goal. This idea refers to the emotional needs and desires of the system users that in many instances, particularly for socio-technical and domestic systems, motivate the existence of system [2], [3]. For example, emotions are the main driving force behind the video game industry, which is worth $67 billion in 2012 [1]. People play video games to have fun, avoid boredom, feel excitement, thrill or fear, among many others.

In our earlier work [3], we distinguish between two types of emotional goals:

1) Personal emotional goals: These represent how people want to feel independent of any system. They often motivate functional goals. For example, I want to feel independent, so I have a goal of buying a car.

2) System-dependent emotional goals: These represent how people want to feel in relation to the system itself; that is, if the system did not exist, the emotional goal would not exist either. For example, when driving a car I want to feel in control of it.

Figures 2 and 3 provide examples of the notation used in our models. Functional goals are represented as parallelograms, quality goals are clouds, emotional goals as hearts, 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.

2) Functional goals to quality goals: this represents that the functional goal should be achieved under consideration of this quality.

3) Functional goals to functional goals: this represents that the functional goals are related. We use undirected
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. As a matter of choice, we put the sub-goal below the goal. This hierarchy is useful for top-down modelling.

4) **Roles to emotional goals to functional goals**: this ternary relationship specifies that the role wants to feel the emotional goal, which is in turn satisfied by the functional goal.

In Miller et al. [3], we discussed these concepts in detail, and propose how to incorporate them into goal models, motivational scenarios and role models – models that motivate the design and development of the system, but do not describe requirements in sufficient detail for development and verification. In this paper, we prescribe additional models that capture system behaviour, and that trace requirements back to emotions. We will illustrate the models using an example from the social media domain: our earlier example of an application to allow people to share pictures with their peers.

B. **Motivational context models**

An important part of understanding the problem to be solved is to describe its context. Often, solutions under development are intended to be integrated into existing socio-technical systems, either computer- or human-based. The motivational context model can be used to describe the situation without the sought system to be developed and the gap that it will fill. This seeks to ensure that once the solution is developed it will not be incompatible with the pre-existing systems. The motivational context model uses the notation of goal models described by Sterling and Taveter [9]. For example, the motivational context model shown in Figure 1 emphasises that the mobile app will support photo sharing, and will complement other existing ways of virtual social interaction, such as microblogging.

The motivational scenario is adapted to include the description of emotions. This model has been discussed in an earlier paper [3], we refer the reader there for further detail.

C. **Emotion models**

Systems are typically defined to solve a problem, or a so-called pain point. In some cases, people express their requirements in terms pain points; e.g. they state what they do NOT want rather than what they do want. In the context of emotional needs, these are often expressed as negative emotions, e.g. *I don’t want to feel irrelevant*.

In requirements engineering, we model what the system must achieve, rather than what it should not, so for backwards traceability, it is useful to map between what the system must achieve, and what problem each part solves.

For this, we introduce emotion models, which map the negative emotions with the functional, quality and emotional goals that the system should achieve to counteract them. All negative emotions should be addressed by some goal or combination of them to ensure that all concerns from users are taken into consideration. However, not all goals must be mapped to negative emotions, as there could be aspects of the system which are not driven by emotional concerns. This model helps developers maintain the traceability between types of goals and the negative emotions that they address. It also serves as a sounding board to remind developers why they are working on certain aspects and to help them prioritise requirements in the event of having restricted resources.

Figure 2 presents an emotion model for the photo sharing example. To represent negative emotions, we use spades, because these are in contrast to hearts in a pack of cards. In Figure 2, the functionality of photo sharing and the emotional goals of feeling engaged and in touch address the negative feelings of feeling bored and isolated, respectively. Similarly, the functionality of photo sharing and the emotional goals of feeling popular and recognised seek to alleviate the feelings of irrelevancy and being ignored.

D. **Goal models**

The goal model is the primary tool to discuss the problem with the stakeholders. This model uses a simple notation to capture the main functional goal of the system, decomposed into sub-goals. It also shows the roles that are responsible for achieving functional goals. Quality goals can be attached to the corresponding functional goals to represent how the functional goal should be achieved. Our modification is to
add emotional goals to the goal model. The addition of the emotional goals is new to the original version of the goal model. The model depicts emotional goals linked to functional goals. In our experience in using emotional goals, we have not found it necessary to distinguish between personal emotional goals and system-dependent emotional goals, as the context is often clear enough. However, in cases where this is not the case, we propose that personal goals are above the corresponding functional goal, and system-dependent goals are below. This has the implicit semantics that personal emotional goals motivate the functionality, whereas the system-dependent emotional goal is dependent on the functional goal.

As an example, consider the goal model depicted by Figure 3. The main functional goal, share photos, is motivated by the emotional needs of feeling engaged, in touch, popular and recognised. The system has to be designed in such way that is immediate and simple to use. The main functional goal is decomposed in several sub-goals, namely add photo, add tag, find photo and comment. The sharer first adds a photo to the sharing platform (the photo should be suitable) and adds tags. Then, their friends and other people find the photo and engage with the sharer in an exchange of shared comments.

In Miller et al. [3], we present a more detailed discussion about goal models, and the relationship between emotional goals and other types of goals in these models.

E. Role models

An essential element to be discussed with stakeholders is the definition of the roles involved in the system. This is facilitated by role models. As these models have been introduced in our earlier paper [3], we will be brief here. The last two rows of Table I illustrate how emotional goals are used to extend Sterling and Taveter’s models [9]. The rest of the fields correspond to the functional goals (Responsibilities) and quality goals (Constraints) as per goal model (Figure 3).

F. Interaction models

Interaction models capture the interactions that occur between roles in a socio-technical system. Sterling and Taveter [9] propose three types of model to represent interactions: (1) interaction diagrams; (2) interaction-sequence diagrams; and (3) interaction protocols. There is a high degree of overlap between these models, so it is likely that developers would choose at most one of them to model interaction, however, each one introduces additional aspects that might be useful in different situations. We refer the reader to [9] for a detailed discussion on the differences and nuances of these models.

Essentially, interaction models specify what types of interactions occur between people/agents playing roles. Our key insight is to annotate these models with emotional goals to indicate at which point of an interaction an emotional goal could be fulfilled.

An interaction diagram (Figure 4) describes a dialogue between agents. It represents the interactions between agents as arrows from and to their life lines. Interaction diagrams do not distinguish between mental or physical actions and do not support alternative interactions, as in UML diagrams. Our extended interaction diagrams link back to emotions, demonstrating points in the interaction at which emotional goals could be fulfilled. This is done by annotating the models with emotional aspects underneath the interactions. The interaction diagram depicted by Figure 4 shows how a sharer adds a photo to the sharing platform. This triggers interactions between the sharer and their friends. In this case, the emotional goal of being recognised is not fulfilled by adding photos, but when other users comment on those photos in a positive manner.

The emotional annotations support straightforward tracing from emotional goals to interactions, raising the system designers’ awareness of the emotional factors that need to be considered when designing the interaction mechanism. Similar annotations can be added to interaction-sequence diagrams and interaction protocols to enable tracing of emotional goals.

G. Scenarios

In Sterling and Taveter’s models, scenarios elaborate the interaction models to include information about resources and quality goals. We extend the definition of scenarios by adding the emotional goals involved in the scenario, and in each step of the scenario. Emotional goals associated to the entire scenario are represented in a row in the table. For each step
in the scenario, emotional goals are annotated in columns. As with the emotional goals in interaction diagrams, this supports traceability of emotion goals back to their source. Table II shows the scenario corresponding to the interaction diagram depicted by Figure 4. Note that the emotional goal Engaged does not appear in the activities, as it is not associated as much with individual interactions as with the overall use of the system.

H. Agent behaviour models

Sterling and Taveter [9] define agent behaviour models to describe the internal behaviour of agents. However, in socio-technical systems, roles are often played by humans as well as artificial agents. To model the behaviour of the system beyond the agents’ internals, we use a statechart-like model, in which the models are annotated to indicate the emotional goals involved in each state, and the roles that trigger the transitions. Much like annotating interaction models, these annotations allow traceability from emotional goals through the requirements. Figure 5 illustrates this model for the picture sharing example.

IV. MODELLING PROCESS

In this section, we present the modelling process that we propose to build the set of models described in Section III. We have designed the modelling process in the same spirit as the models. We are merely describing, based on our experiences, a process that has worked for us, and which we believe represents a model that can be adopted by others. We describe a complete process to produce all models, however, we are not dogmatic about its application, nor prescriptive in excess. In particular, it is unlikely that all of the models would be required in any single project. We encourage others to understand the expressive power of each model and use those that are most suitable for any particular situation.

The modelling process is rooted on the belief that understanding the problem is essential to solve it adequately. The process, therefore, focuses on the requirements engineering and early design phases of software development. The set of models produced at its conclusion could be used as basis for making detailed design decisions and proceed with the implementation of the system. Figure 6 sets the context of our modelling process in relation to other phases of software development in a classic lifecycle. However, any other development model, such as agile practices, could also benefit from the application of using such as lightweight modelling process. The modelling process, as a whole, requires knowledge the problem to be solved as an input. Acquiring this knowledge is out of the scope of this paper and we refer the reader to the abundant literature on elicitation techniques. In our experiences we have adopted a number of approaches, including ethnographic studies, interviews or brainstorming sessions [3], [5], [6].

Figure 7 presents the process model, which we believe is natural based on the information modelled in each type of model. Some models can be built at any time and some models are dependent on others. We build our models using type
<table>
<thead>
<tr>
<th>Scenario</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Functional Goals</strong></td>
<td>Share photo</td>
</tr>
<tr>
<td><strong>Quality Goals</strong></td>
<td>Immediate, simple, suitable</td>
</tr>
<tr>
<td><strong>Emotional Goals</strong></td>
<td>Engaged, popular, recognised</td>
</tr>
<tr>
<td><strong>Initiator</strong></td>
<td>Sharer</td>
</tr>
<tr>
<td><strong>Trigger</strong></td>
<td>The sharer wants to share a photo</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Condition</th>
<th>Step</th>
<th>Activity</th>
<th>Roles</th>
<th>Resources</th>
<th>Quality Goals</th>
<th>Emotional Goals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential</td>
<td>1</td>
<td>The sharer shares a photo</td>
<td>Sharer</td>
<td>—</td>
<td>Immediate, simple, suitable</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>The sharer adds tags to the photo</td>
<td>Sharer</td>
<td>—</td>
<td>Immediate, simple</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Other system users find the photo</td>
<td>Friend, Peer</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Friends and peers comment on the photo to express admiration</td>
<td>Friend, Peer</td>
<td>—</td>
<td>—</td>
<td>Popular, recognised</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>The sharer replies to the comments on the photo to express their gratitude</td>
<td>Sharer</td>
<td>—</td>
<td>—</td>
<td>In touch</td>
</tr>
</tbody>
</table>

setting and drawing tools, however, in some situations, such as agile development meetings or discussions with stakeholders, we take printed versions on large paper and ask stakeholders to modify models directly.

The modelling process starts with the *motivational context model*, as it does not focus on modelling the problem itself, but its context. This model does not describe any aspect that will be used in other models explicitly, but uses domain knowledge to share the context in which the system will operate.

The *motivational scenario* is the first model that considers the system properties. It does not depend on the motivational context model, but establishing the boundaries of the socio-technical system may provide benefit at this stage. This model takes the domain knowledge to describe why the system is to be built, in terms of what functionalities, qualities and emotions are to be achieved.

The *emotion model* organises all aspects identified in the previous model around the negative emotions that are targeted to be mitigated with the creation of the system. This model will allow developers to keep the perspective, i.e. to trace goals to their motivation, hence, helping with prioritisation.

The *goal model* organises and structures the functional, quality and emotional goals listed in the emotional model. Functional goals are structured in a hierarchical way and quality and emotional goals are associated with the functional goals that they modify. Based on domain knowledge, stick figures representing goals are associated to functional and emotional goals. We have found that goal models are particularly useful to share with stakeholders early in a project. The simple notation makes the models palatable for non-technical stakeholders. We advocate that developers brainstorm with stakeholders about the system to be developed using a goal model as a boundary object. We have also found that printing large versions of the goal model and bringing markers to the meeting to allow stakeholders to directly modify aspects of the model instills in users a sense of ownership over the final product [4].

The roles initially identified in the goal model are elaborated in the *role model*. The relationships established in the goal model between roles and functional and emotional goals are formalised here as well. The role model also includes as constraints any quality goal linked to the functional goals pursued by the role.

Once the roles have been properly defined in the role model, *organisation, domain and agent models* can be created. They are based on the roles definition to establish the control relationships between them, their use of environment entities – based on the domain knowledge – and to map roles to agents. As these models have no strict dependencies among themselves, they can be developed in any order or concurrently. The agent model is the first model that pushes modellers to make early design decisions; i.e. the mapping between system roles and agents. From this point in the modelling process onwards, all models require making design decisions.

Completing the agent model enables the creation of the *acquaintance and knowledge models*. Both models complete the definition of agents by describing the interaction channels between them and by detailing what knowledge – in terms of data types – agents require to perform their allocated tasks.

The *interaction models* describe in detail the interactions between the agents in the system. They are based in the acquaintance model. Any of the three types of interaction models can be build at this stage, namely *interaction diagram*, *interaction sequence diagram* and *interaction protocol*. We find that even simple interaction diagrams provide effective guidance to implement the system in later development stages,
Fig. 7. Suggested modelling order

particularly for interaction-intensive systems.

The scenario models require the interaction model and the domain model. Looking at the interactions described in the former, the modeller will be able to synthesise the activities that have to be completed to achieve the system goals, in which order, what agents will be involved in those activities and what quality and emotional goals will be associated to them. From the domain model, the modeller will compile the resources required by the activities.

The behaviour model can be developed in parallel with the scenario models. Both types of model are independent from each other and both depend on the completion of the interaction model. The behaviour model describes the system interactions from the point of view of the agents in terms of triggering rules and messages. This model and the service model bridge the initial design of the system with more detailed design decisions. The service model is based on the behaviour and domain models to express the exchanged messages and environment services as invocations that could be directly translated into programming language specific statements.

To decide what models will be built, modellers should carefully consider how the final system will be implemented. Depending on the specific application, some models will be more helpful, or other models will have to be modified to fit particular modelling and development styles.

V. EXPERIENCE

We believe that by explicitly modelling emotions we can improve systems by identifying requirements that are unlikely to be uncovered otherwise. To test this, we used the models that we have described in Section III and the process presented in Section IV to build a tablet app in the domain of personal alarm systems for older adults. In this section we describe the problem domain and the experience.

A. Personal alarms for older adults

The use of personal alarms is common amongst older adults that want to remain living at home independently. These alarms are typically composed of a fixed base station connected to the landline, and a mobile device, usually in the form of a pendant, that can be used to call for help within the base station range. A constraint of the system is that the older person must have the pendant within reach at all times to raise an alarm when necessary. A feature that service providers often bundle with the personal alarm is a wellbeing check. This service consists of a button, often positioned on the base station, that has to be pressed daily to indicate that everything is okay. If the button is not pressed within a defined time frame established by the service provider, the older person is contacted via phone to check whether they need help. If the service provider cannot reach the person, their nominated contacts (e.g. family members) are notified. In case that none of the contacts is available, the emergency services are alerted.

We became interested in this case study because most of us knew older people who used personal alarm systems and generally disliked them, despite the alarms functioning reliably. If these feelings are so common amongst users, why do service providers not improve them? The answer was that their surveys consistently showed an extremely high satisfaction among the system users. In fact, these systems are excellent from a technical and engineering point of view: they are reliable and robust.

Our hypothesis as to why there is such a disconnect between our anecdotal evidence and the survey data of emergency alarm service providers was simple: the surveys were asking the “wrong” questions. By only asking users about how well the functional aspects of the system operated, they were not getting the complete picture. We undertook a study to ask the older people and their relatives about the system, including questions about their emotional perceptions. The data revealed a different story to that of emergency alarm manufacturers and service providers. Many of the older persons that have a personal alarm dislike it and do not use it properly. In some cases, this misuse has dire consequences. In one instance, one older person opted to never wear her pendant, and subsequently fell
in her home and lay on the floor 36 hours until her relative checked on her. The study provided evidence that the older people’s emotions played a determinant role in the rejection of the system. A prevalent feeling was that the pendant made many older people feel stigmatised and less independent. Some older people would wear the pendant only when they knew their relatives would come for a visit and would put it away in a drawer as soon as they were alone. Because of this, cases of older people falling at home while being alone and not being able to call for help were, were not uncommon. In the case of the wellbeing check, many older people failed to appreciate this at all, often forgetting to push it, and in many cases, having no idea why they were pushing it or what it did — they were simply told: push the button every day. We believe that by considering the feelings and emotional needs of the older adults during the design and development of the system, the resulting system would have been much better accepted.

B. Redesigning the “wellbeing check” for emergency alarm systems

Based on this preliminary study, we tackled the development of a system to implement the wellbeing check functionality of personal alarm systems in a emotionally-oriented way. We prefer to call this aspect of the system *I’m in touch*, as it reminds the developers that the emotional need of the older person is to keep in touch with their friends and relatives; not to “check in” with a third party service provider. In this preliminary system, we did not offer an alternative to the pendant due to the high risk involved.

We followed the process described in Section IV to develop the complete set of models for this problem. Figure 8 shows an excerpt of the goal model for the *in touch* functionality of the system. The emotional goals are propagated to the appropriate models (Section III), for example the interaction protocol depicted by Figure 9.

The main aspects that we felt have been understated in current systems were the quality and emotional goals shown in the model depicted by Figure 8. An important quality aspect is that the system should be part of the routine of the older adults, they should not have to deviate from their everyday life to push a button, because that is prone to memory lapses. By communicating with the older adult, their carers and relatives seek to feel reassured that the older person is okay. By using the system, the older adults want to feel cared about (not cared for – a significant finding), independent and in touch. They also want to feel that the system is integrated in their lives and that it is under their control. Both parties want to feel unburdened from a compulsory and sometimes meaningless daily check-in call.

We translated these functional, quality and emotional goals to the design. The resulting application is an interactive picture frame that runs on a tablet. It could be used as a communication means between older people and their carers/relatives. The carer/relative would send a picture that they hoped the older person would like. This picture would be shown on the older person’s photo frame and they would be...
able to make comments or just view it, establishing a two-way
communication. Every time the older person moved the photo
frame, change the photo being displayed in the photo frame,
or commented on a photo, the system would interpret this
activity as a signal that the person was okay. By designing the
system in such way, we provided the older adult a mechanism
that was integrated in their life, as the picture frame would
sit permanently on display. Because the pictures sent were
chosen by their carer/relative, they could be meaningful and
would make them feel cared about. By exchanging comments
with the carer/relative, the older person would feel in touch
and the carer/relative reassured. The photo frame is simple
and designed to give immediate feedback to the older person
during its operation, so it would contribute to the feeling of
being under the control of its user. By providing asynchronous
and meaningful communication, the application supports the
feeling of being unburdened.

C. Results

Among the models described in Section III, the goal model
was the most useful in this project because it compiles most
information needed to understand and motivate the problem.
Due to the simplicity of its notation it is a good tool to
mediate the discussion with stakeholders, including project
participants. It is lightweight and can be sketched together
with the stakeholders to agree on ideas before producing more
specific requirements; for example, in the form of user stories.
This echoes the findings from our previous experience [3]. We
believe that the benefits of creating a goal model and refining
it according to the stakeholders’ input outweigh the overhead
that is introduced. Besides the improved agreement between
developers and stakeholders on the problem to be solved, this
model can also be used internally by the development team for
project management purposes. Adding a colour codification to
the goal model has the benefit of being simple to implement
and giving intuitive visual cues on the overall progress of the
from red to green to codify the progress of goals. The goal
models are then printed in large posters and kept on the wall
of the workshop for all developers to see at a glance the
overall progress of the project. Also, this was used for project
management purposes, as developers could be moved to parts
of the project that were lagging behind.

In Section IV we suggest choosing models more appropriate
for a project. For our case study, we produced the complete set
of models to evaluate which were most useful. In hindsight, we
can identify the models that we found more useful in our par-
ticular case study. The two models that were most useful were
the goal model (as discussed above) and the role model. The
latter was helpful to define the roles that were needed in the
system and the boundaries for their responsibilities. We also
found that the interaction models were helpful. Particularly,
the interaction diagram was the simplest to create and was
sufficiently expressive for our case study. Not as useful, but
still worth developing, were the motivational scenario, context
model and emotion model. These models helped establishing
the initial descriptions and limits of the project and were
good at motivating the “why” questions. In general, we found
the earlier models more useful than the later ones that make
design decisions. We believe that this is partially because our
developer based heavily on the motivation and analysis models
to understand the problem and then he made his own design
decisions. As such, models such as the knowledge model,
behaviour model or service model were mostly overlooked.
Nevertheless, we would expect that in projects where the same
team is involved from the motivation phase throughout to the
implementation of the system, the traceability provided across
models would prove very valuable.

An interesting discussion that needs to occur at the be-
ginning of the modelling process involves the context moti-
vational model. This model describes the situation prior the
development of the system in terms of goals and roles. In
this model, we chose not to include the pre-existing emotions.
We did not add these because they will be elaborated later
in the emotional model. However, this model presents a good
opportunity to discuss the emotional boundaries of the larger
socio-technical system; i.e. whose emotions will the system
cater for? The answer to this is domain specific, but in the
case study presented in this paper, we wanted to develop a
system to address the emotional needs of the older people as
well as their relatives and carers, who are often drivers of
installing these systems. We are aware that neighbours will
also feel concerned about the wellbeing of the older person.
Even the employees of the service provider that help the older
person weekly with their tasks may develop a bond and feel
concerned about them. However, these emotions were not the
focus of the system.

We will be trialling this new application over several months
in late 2014. A shorter trial using a prototype version of this
application has already been trialled [3], [6]. We installed the
prototype in several older people’s homes who volunteered to
trial it. The analysis of the data obtained from this experiment
(in the form of interviews and interaction logs) shows an
improved user experience over traditional “wellbeing checks”,
which we attribute to the prototype considering important
emotional needs of older people. Consequently, they have
more positive feelings using the system and feel cared about
instead of patronised and monitored. This short trial is dis-
cussed in detail in [6].

VI. Discussion

During our project we have read extensively on emotions
in the psychology literature. We compared a number of clas-
sifications trying to find one that we could use to classify the
types of emotions that we expected to encounter in software
and systems development projects. However, we could not find
an adequate resource. For example the emotions of feeling in
control, cared about, or feeling that the system is integrated
in your life, did not fit with psychology literature, which
focused on classical emotions like joy, guilt, embarrassment,
etc. Nevertheless, we felt that these emotional goals were valid
and real emotions that had to be considered in our case studies.
We decided to fully embrace the pragmatic approach that we were using in designing our models and avoid trying to find a framework for emotion classification, which may not be flexible enough.

The notation for emotions that we have adopted in this paper, as hearts, was pioneered by Marshall [2], who first suggested that emotional goals should be treated separately from quality goals. In previous work [5], we had modelled these social goals as quality goals, but discussions with Marshall convinced us that we should treat these separately as emotions. The notation for negative emotions that occur in the system and have to be dealt with, is inspired by the similarities with card games. The emotional goals are positive emotions, represented as hearts. The negative emotions to be avoided are represented as spades, which are black and have negative connotations.

It is worth noting, that using our agent-oriented models does not necessarily imply that an agent-oriented implementation should follow. For example, the first layer of abstraction defined by Sterling and Taveter [9], the motivational layer, delimits and motivates the problem. The models defined in this abstraction layer are useful to better understand the problem and as stepping stone to build a solution using any other technology; for example a mobile application following an agile development process. Indeed, because of their simplicity and ease of understanding, they could be used in agile developments to improve the resulting product without adding an excessive overhead [10].

VII. CONCLUSION AND FUTURE WORK

In this paper we propose an extension to a requirements engineering methodology to adapt it to the new social age by focusing on emotions. We believe that many socio-technical systems present a low level of acceptance because they fail to cater for the emotional needs of their users. We suggest incorporating the emotions that people feel (or want to feel) into requirements engineering activities. We build on existing works to extend agent-oriented models and we define a process model. The emotional goals can be traced throughout the requirements, including their source. We illustrate our proposal using the example of a case study of personal alarm system for older people that live independently at home. By applying our process, we produced an application that we believe caters better for the emotional needs of older adults, which we will be trialing later this year.

As our next step, we will undertake formal evaluations of our requirements engineering approach, and will continue to evaluate it in different domains, including domains such as insomnia, depression and psychosis. In parallel, we are studying how to formalise the transition of emotional goals into the design process to produce a solution that caters for the emotional needs of stakeholders. We believe that an important element of this problem will be to analyse how people with different personal values interpret the emotional goals. Finally, we are working on improved systems for supporting wellbeing of older people at home.

VIII. ACKNOWLEDGEMENTS

This research has been funded by the Australian Research Council (ARC) Discovery Grant Software engineering of human-oriented technologies and services (DP1301022660).

REFERENCES