Eliciting and visualising trust expectations using persona trust characteristics and goal models.

FAILY, S. and FLÉCHAIS, I.

2014

Eliciting and Visualising Trust Expectations using Persona Trust Characteristics and Goal Models

Shamal Faily
Software Systems Research Centre
Bournemouth University
Poole, UK
sfaily@bournemouth.ac.uk

Ivan Fléchais
Department of Computer Science
University of Oxford
Oxford, UK
ivan.flechais@cs.ox.ac.uk

ABSTRACT

Developers and users rely on trust to simplify complexity when building and using software. Unfortunately, the invisibility of trust and the richness of a system's context of use means that factors influencing trust are difficult to see, and assessing its implications before a system is built is complex and time-consuming. This paper presents an approach for eliciting and visualising differences between trust expectations using persona cases, goal models, and complementary tool support. We evaluate our approach by using it to identify misplaced trust expectations in a software infrastructure by its users and application developers.

Categories and Subject Descriptors

D.2.2 [**Software Engineering**]: Design Tools and Techniques—*Computer-aided software engineering (CASE)*

General Terms

Design

Keywords

Personas, Goal Models, i*, GRL, Trust, CAIRIS

1. INTRODUCTION

It has long been acknowledged that, for systems to be secure, they must also be usable. For a system to be usable, it must attend to the contexts of use associated with all of its users; these include their goals, the artefacts they use, activities they undertake and behaviours they exhibit, and the physical and social environments within which people operate. The ability to compute the consequences of the myriad of social interactions between people in these contexts of use is an important element of social computing [2]. However, the richness of contexts of use mean that conflicting goals arising from different user expectations may be difficult to identify while a system is still being built.

Trust is a pervasive element in systems that need to support social behaviour. Developers rely on trust to reduce complexity when building a system, while users rely on trust to remove any unnecessary cognitive burden when using the same system to satisfy their goals. However, trust and trustworthiness mean different things to different people. For a given activity, individuals with the same role may have different personal characteristics, and subscribe to different norms about how an activity might be undertaken, and about the social context in general. If the activity is critical then divergences which seem innocuous to one person may have a catastrophic impact on somebody else. Consequently, some means of assessing whether a given combination of people, activities and goals meets specified trust expectations would be invaluable.

The complexity of social contexts means the ability to carry out some form of trust assessment is time-consuming without software support. Building such tools entails the development and alignment of different models encapsulating the knowledge needed for such an assessment. While the types of models needed to reason about the trustworthiness of social agents within different contexts appear to be eclectic, much of the related work necessary to facilitate alignment between models already exists. To illustrate this, we present an approach for eliciting and visualising differences between trust expectations using persona cases, goal models, and complementary tool support. We describe the related work grounding this approach in Section 2, before presenting the approach itself in Section 3. We describe a case study evaluating the approach in Section 4, before discussing some limitations and consequences of this work in Sections 5 and 6 respectively.

2. RELATED WORK

2.1 Trust and Trust Properties

Trust is a social norm that can be considered as the willingness to be vulnerable, based on the positive expectation about the actions of others [29]. This expectation can be tied to many factors, and the mediating role of technology plays its part in altering or removing signals that a person or system might rely on to establish trustworthiness. To explore these factors, Riegelsberger et al. [24] developed a framework based on the sequential interaction between *trustors* (trusting actors) and *trustees* (trusted actors). In this framework, trust is the trustor's internal state concerning the expected behaviour of the trustee in the given context.

Riegelsberger's framework identified several *intrinsic* and *contextual* trust-warranting properties. Intrinsic properties are attributes of a trustor or a trustee that lead them to behave in a trustworthy manner; examples of such attributes include motivation (what the actors gain from being trustworthy), ability (their inherent competence at carrying out the activity), internalised norms (their beliefs, values, and usual patterns of behaviour), and benevolence (the gratification earned from helping another). Contextual properties are attributes of the context that motivate trustworthy behaviour; these include temporal embeddedness (how trustworthy behaviour will influence future interactions), social embeddedness (how trustworthy behaviour will influence interactions within a social grouping), and institutional embeddedness (how trustworthy behaviour relates to the broader organisational context).

This framework has been used to reflect on the relationship between trust and related security concepts such as reliance and assurance [16]. For example, Fléchais et al. explain that internalised norms can induce social actors to behave in a trustworthy manner, and analysing these can indicate whether actors are likely to break trust.

2.2 Modelling Trust with i*

Work by the Requirements Engineering community has demonstrated how the norms alluded to by [16], and the socio-technical elements associated with them, can be visualised using i* (Intention STrategic Actor Relations): an agent-oriented approach for modelling and analysing stakeholder interests, and how they might be addressed or compromised in various system and environment alternatives [28]. Certain concepts are key to i*:

- *Goals* are conditions in the world that stakeholders would like to satisfy.
- *Softgoals* are goals with ill-defined satisfaction criteria. Such goals are satisficed (as opposed to satisfied) if they are achieved to an acceptable degree.
- *Actors* represent autonomous agents (people, hardware, or software) with intentional goals.
- *Tasks* represent specific procedures performed by actors.
- *Resources* are physical or informational entities.

If an actor intends to achieve a goal then this can be achieved by either carrying out a task, or by relying on another actor for the achievement of that goal, or a resource or task necessary for an actor to achieve the goal. These socio-technical system elements are modelled in two types of model. *Strategic Dependency* models illustrate the goal, task, or resource dependencies between actors. *Strategic Rationale* models explain how goals or softgoals positively or negatively contribute to the achievement of other goals, softgoals and tasks; they also connect an actor's intentional elements to strategic dependencies of other actors, thereby connecting to strategic dependency models.

In recent years, i* has formed the basis of the Goal-Oriented Requirements Language (GRL): an international standard for uncovering, analysing, and describing stakeholder goals [1]. GRL shares many concepts with i*, but does not distinguish between separate strategic rationale and dependency models. GRL is tool-supported by jUCMNav: a graphical editor that supports the creation and assessment of Goal-oriented Requirements Language (GRL) models [22]. GRL models are assessed by creating *strategies*; these indicate the satisfaction of model elements based on the initial satisfaction level of one or more elements, and the contribution links between them.

Unfortunately, i* based languages have yet to reach their potential due to the difficulty creating and using them, and the languages' bias towards modelling rather than the elicitation and analysis of model data. An evaluation of the visual effectiveness of i* [21] found that its graphical complexity – the number of different elements used in the visual notation – is nearly three times greater than a human's standard limit for distinguishing alternatives; this complexity may significantly reduce the understanding of models especially by novices.

Despite their challenging nature, several attempts have been made to use i* based languages to model the impact of trust. For example, Elahi and Yu [7] propose annotating i* dependency relationships with 'trust rationale'. Similarly, Mouratidis & Giorgini have considered how trust might be incorporated into Secure Tropos: a security-oriented extension to the Tropos method, which also adopts the i* framework [19]. This extension entails the introduction of ownership, trust, and delegation concepts, together with processes for modelling and formally analysing trust and delegation chains.

Such trust extensions confound the problems highlighted because both [7] and [19] expand and, in the case of [19], overload existing i* elements. Problems have also been found reconciling models in line with stakeholder views, identifying the most appropriate place to start modelling, and modeller confusion between cognitive trust and trustworthiness [20, 23].

2.3 Evaluating the Impact of Trust with Personas

Like the modelling frameworks described in Section 2.2, usability models describe the impact of human factors when imagining how a system might be used. Such models are widely used by User Experience (UX) practitioners, and UX techniques like scenarios have been successfully appropriated by software engineers for several years. One such technique of growing interest to software engineers is the *persona* technique. Personas are narrative descriptions of archetypical users that embody their goals and needs [4]. To contextualise them, personas are often complemented with scenarios; these envision how potential users (represented as personas) might interact with the system within its intended or unintended context of use.

Because they are comparatively easy to develop and use, personas are becoming popular for summarising user research about prospective system stakeholders [3]. Personas have also been proven useful when engaging stakeholders in security [10]. The activities necessary to create and apply personas are also conducive to a security analysis because as well as identifying affordances for use, affordances for misuse and possible vulnerabilities can be identified at the same time [13]. For example, as a means of simultaneously contextualising sceanarios and envisioning system vulnerabilities, *security premortem* scenarios can be applied. These are scenarios presented to stakeholders that assume a system has

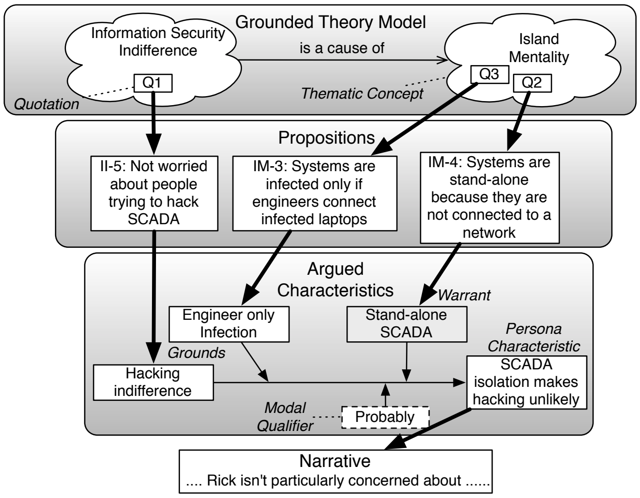

Grounded Theory Model
Information Security Indifference
is a cause of
Island Mentality
Q1
Q3
Q2
Quotation
Thematic Concept
Propositions
II-5: Not worried about people trying to hack SCADA
IM-3: Systems are infected only if engineers connect infected laptops
IM-4: Systems are stand-alone because they are not connected to a network
Argued Characteristics
Warrant
Engineer only Infection
Stand-alone SCADA
Persona Characteristic
Grounds
Hacking indifference
SCADA isolation makes hacking unlikely
Modal Qualifier
Probably
Narrative
.... Rick isn't particularly concerned about ......

Figure 1: Deriving a persona characteristic from a grounded theory model relationship (taken from [12])

been exploited, who are then invited to present plausible reasons are given for explaining why [15]. The premortem scenario sensitises stakeholders to the personas and the relationship between system goals and persona expectations. If used effectively, these scenarios can lead to the generation of insights that might otherwise be missed when considering personas and their normal course scenarios alone.

While personas and scenarios are useful for reflecting on the typical behaviour of target users, they are less useful when considering less frequent behaviour of trust import. However, work on the Persona Case framework has shown that qualitative models, such as those that conceptualise trust, can be used to derive personas [12]. This involves treating a design problem as a research problem, and devising research questions to characterise it. Qualitative research is then conducted to collect empirical data from the user population of concern; this might include running qualitative interviews, or some form of ethnographic study. The qualitative data collected is then coded and analysed using Grounded Theory [5] to develop a conceptual model that tackles the research problem. Using argumentation models, each relationship from this conceptual model is structured to motivate and justify a persona's characteristic. This is summarised in Figure 1.

Personas derived from the Persona Case framework consider the impact of trust on their characteristics, but this impact is largely divorced from systems they might interact with. It has, however, been shown that, if an activity can be characterised in use cases, and actors participating in these use cases can be characterised as personas, then use case and persona case elements can be aligned with complementary elements of GRL [8]. This alignment makes it possible to not only create models that contextualise the impact of personas on socio-technical system, but also the influence of trust on these personas.

3. APPROACH

We have devised an approach for exploring how warranted the trust expectations of different system stakeholders might be. The Persona Case framework [12] is used to develop a qualitative model from which trust characteristics of personas are derived. By re-framing these characteristics and their supporting elements as social goals and contribution links, GRL models can be generated. These models are then subject to further analysis to explore additional dependencies between personas. We elaborate this approach in the sub-sections below.

3.1 Eliciting Persona Trust Characteristics

The first step involves eliciting persona characteristics that attend to trust expectations. To do this, we use Riegelsberger's framework for trust in technology-mediated interactions [24] to support a grounded theory analysis of trust factors influencing a particular user population. The research questions that drive the coding process are motivated by this framework, and include questions about intrinsic and contextual trust properties. We then use the Persona Case framework to derive intrinsic or contextual persona characteristics from each relationship in the grounded theory model.

3.2 Deriving GRL Model Elements from Persona Characteristics

Once the persona characteristics have been created, we analyse both the characteristics and their contributing elements to identify goals, softgoals, or tasks that might be implied by the characteristic. For each contributing element, we also identify whether the element is a 'means' or an 'end' given the contribution relationship, and how much the element contributes to this relationship. More information about how persona characteristics align with these GRL elements is provided by [8].

3.3 Generating and Analysing the GRL Model

A GRL model is now generated by CAIRIS to visualise the social goals associated with the personas in a given context of use, as well as the contribution and dependency relationships between them. The model is created by exporting the GRL elements associated with a particular activity. The GRL model created is then directly imported into jUCMNav where, in conjunction with system stakeholders, it is interpreted in two steps. In the first step, strategies are applied to explore the implications of dissatisfying particular goals, softgoals, and tasks. These strategies also help identify contribution links between persona goals which appear to be missing. In the second step, possible dependencies between the personas in the GRL model are identified using security premortem scenarios. These scenarios sensitises stakeholders to the personas and their characteristics, and creates context for denied goals. Based on the responses given, the model is evaluated to determine whether such a scenario might be possible and, if so, what dependencies need to hold between the personas to realise it.

Based on any changes made to the GRL model, the qualitative models, personas, and system specifications are revisited based on any new insights.

3.4 Tool Support

Rather than developing software tools from scratch, our framework is supported by customising and building interfaces between two pre-existing software tools: CAIRIS, and jUCMNav.

3.4.1 CAIRIS

CAIRIS (Computer Aided Integration of Requirements and Information Security) is an open-source Requirements Management tool designed to support the specification of interactive secure systems [9]. In addition to managing requirements models such as use cases and goal models, CAIRIS also manages the data associated with several security and usability engineering models, such as risks, scenarios, and personas. CAIRIS complements the use of specific security, requirements, and usability engineering techniques. If these techniques are properly applied, then models arising from them can be directly entered or imported into CAIRIS. The impact that the presence of personas might have on a system's design can then be analysed. For example, [11] illustrates how the alignment between personas, scenarios, and risk analysis models can be used to visualise both the security and usability impact of different scenarios to personas.

CAIRIS was recently extended to support the Persona Case framework set out in [12], and incorporates the functionality necessary to support a grounded theory analysis. This includes the ability to annotate imported text documents with codes and memos, and the functionality necessary for creating and visualising relationships between codes. These extensions are described in more detail in [14].

Given the alignment between models supported by CAIRIS and GRL [8], we have also updated the user interfaces for working with persona qualitative models to include controls for specifying intentional and contribution data associated with quotations. Using CAIRIS' export functionality, it is then possible to generate an XML-based GRL model for a selected activity. This activity needs to encompass one or more use cases and personas.

3.4.2 jUCMNav

jUCMNav is used to visualise the GRL models created by CAIRIS. Using jUCMNav's strategy functionality, it is possible to visualise the contribution of satisfying or dissatisfying one or more goals, softgoals, or tasks in the broader GRL model.

No changes were made to the default installation of jUCMNav to support this approach.

4. CASE STUDY

4.1 webinos

We evaluated our approach by using it to help evaluate the design of a software infrastructure. We were specifically interested in finding potential problems resulting from misplaced trust expectations by its users and application developers. This software infrastructure, *webinos*, is a federated, open-source communications platform. webinos was designed to support web applications running consistently and securely across mobile, PC, home media, and in-car systems [18]. We chose a case study example based on webinos because, unlike many open-source projects, much of the design data upon which the webinos architecture is based is publicly available [26]; this data includes personas that motivated the webinos platform, and use cases specifying how users should interact with applications using webinos.

Identifying trust problems for webinos is difficult for several reasons. First, because infrastructures like webinos are invisible to most users, and perceived only through software applications built upon them [6], any evaluation of the infrastructure will be episodic and based on surface interactions, rather than their implications. Second, while the personas created for webinos were designed to explore the security expectations of target users, the data upon which they were based was not based on security decision making. This means that assumptions need to be made about how personas may think or respond to issues relating to trust. Finally, the invisibility of webinos means that what constitutes effective and secure use is unclear to both users and application developers. For example, users of webinos applications may trust developers to select the most appropriate security defaults for them. Similarly, application developers may be over-confident of their users' ability to configure and use their apps in a secure manner.

We used the approach to identify trust problems associated with installing and configuring the webinos concept application 'Kids in Focus': an in-car game for children. The application was used by a user persona (Helen) and created by an application developer persona (Jimmy); these personas are described in more detail in [27]. The game allows Helen's young son to play an online card game with Helen's father at home. This application was developed by a small team of developers and user interface designers to demonstrate how webinos can facilitate secure communications between an in-car telematics system and a home network. This team was supported by the authors, who were involved in both the human-centered and architectural design of webinos. The source code for *Kids in Focus* is available on GitHub [25].

The specification for installing webinos applications is defined in a single use case (*Installation and update of webinos applications*).

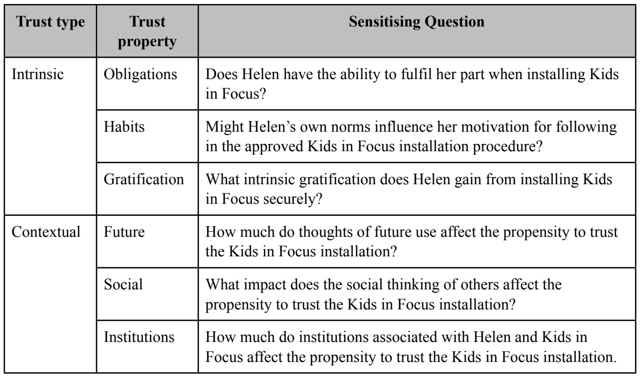
4.2 Approach Applied

4.2.1 Eliciting Persona Trust Characteristics

We began by developing trust characteristics to augment the pre-existing Helen and Jimmy personas introduced in Section 4.1. To do this, we carried out the qualitative data analysis stages of this step using pre-existing data collected during workshops attended by prospective webinos users and developers. These users and developers were recruited based on shared characteristics with the Helen and Jimmy personas respectively, and the workshops considered how the participants made access control decisions. For each persona, three 1.5 hour workshops were held and, following each, the workshop facilitator wrote a report on the outcome of the session.

These reports were subject to a grounded theory analysis to develop trust characteristics for the two personas. Riegelsberger's framework was used to develop a series of *sensitising* questions; these questions help analysts variate and make connections between different codes during the grounded theory analysis [5]. To illustrate these, Figure 2 shows the sensitising questions used to code Helen workshop reports.

From the grounded theory analysis, 57 and 37 quotations were elicited based on the Helen and Jimmy focus group reports respectively; these quotations were based on 26 codes. The relationships between these codes for Jimmy are illustrated in Figure 3; the numbers shown in each box after the



Trust type	Trust property	Sensitising Question
Intrinsic	Obligations	Does Helen have the ability to fulfil her part when installing Kids in Focus?
	Habits	Might Helen's own norms influence her motivation for following in the approved Kids in Focus installation procedure?
	Gratification	What intrinsic gratification does Helen gain from installing Kids in Focus securely?
Contextual	Future	How much do thoughts of future use affect the propensity to trust the Kids in Focus installation?
	Social	What impact does the social thinking of others affect the propensity to trust the Kids in Focus installation?
	Institutions	How much do institutions associated with Helen and Kids in Focus affect the propensity to trust the Kids in Focus installation.

Figure 2: Helen Sensitising Questions

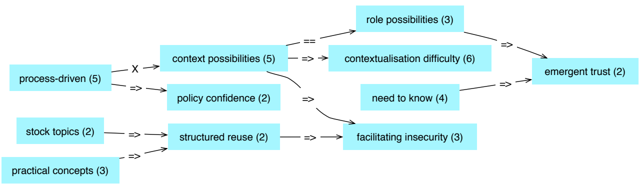


Figure 3: Qualitative model of Jimmy trust characteristics

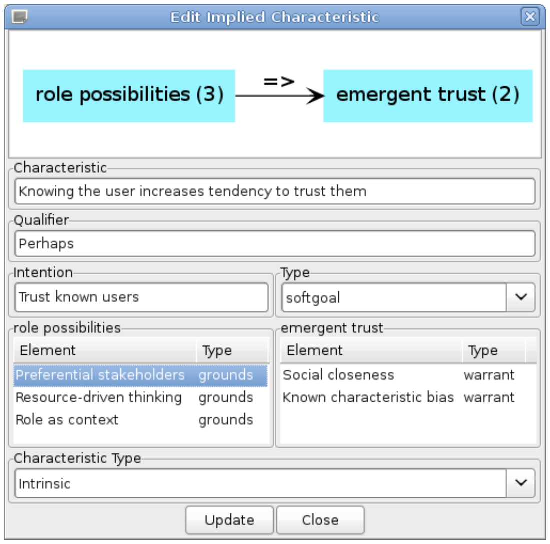
code names indicate how many quotations were associated with that code. Figure 4 illustrates how the characteristic *Knowing the user increases tendency to trust them* is a synopsis of the code relationship between the *role possibility* and *emergent trust* codes.

4.2.2 *Deriving GRL Model Elements from Persona Characteristics*

As prescribed by Section 3.2, the grounded theory relationship behind each persona characteristic was analysed to identify implicit intentional elements and contribution links. In considering the relationship illustrated in Figure 4, the trust that Jimmy develops as a result of knowing his user community implies that he possesses the softgoal *Trust known users.*

Each element underpinning this relationship also suggests implied goals or softgoals that either contribute to this softgoal, or this softgoal makes a contribution to. For example, the *preferential stakeholder* element is associated with the synopsis that security discussions favour one group of stakeholders over others; this implies that Jimmy holds the softgoal *Favour users.* Similarly, *Resource-driven thinking* indicates that developer thinking about security drifts to thinking about resources needed by users, rather than what the system currently offers them; this implies that Jimmy possesses the goal *Determining user resources.* When considering the softgoal and goal within the broader context of the persona, *Favour user* appears to have a significant positive contribution towards the *Trust known users* softgoal, and *Trust known users* appears to have a weak, but nonetheless positive, contribution towards the goal *Determine user resources.*

4.2.3 *Generating and Analysing the GRL Model*



Characteristic: Knowing the user increases tendency to trust them

Qualifier: Perhaps

Intention: Trust known users

Type: softgoal

role possibilities

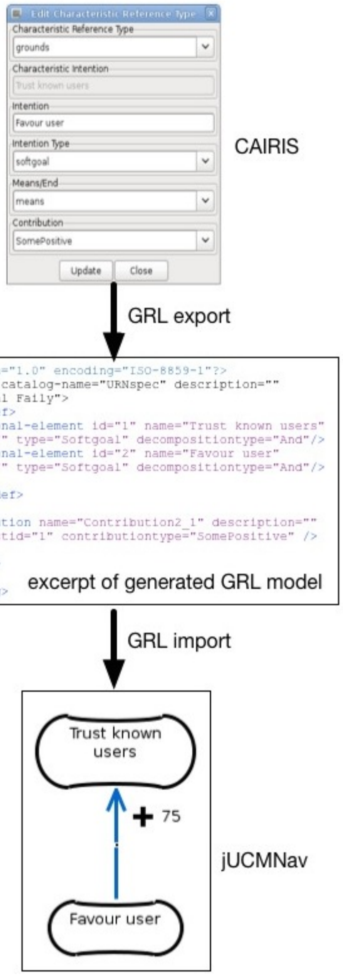
Element	Type
Preferential stakeholders	grounds
Resource-driven thinking	grounds
Role as context	grounds

emergent trust

Element	Type
Social closeness	warrant
Known characteristic bias	warrant

Characteristic Type: Intrinsic

Figure 4: Trust characteristic for Jimmy derived from code relationship



Characteristic Reference Type: grounds

Characteristic Intention: Trust known users

Intention: Favour user

Intention Type: softgoal

Means/End: means

Contribution: SomePositive

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<grl-catalog catalog-name="URNspec" description=""
author="Shamal Faily">
  <element-def>
    <intentional-element id="1" name="Trust known users"
description="" type="Softgoal" decompositiontype="And"/>
    <intentional-element id="2" name="Favour user"
description="" type="Softgoal" decompositiontype="And"/>
    .....
  </element-def>
  <link-def>
    <contribution name="Contribution2_1" description=""
srcid="2" destid="1" contributiontype="SomePositive" />
    .....
  </link-def>
  ....
</grl-catalog>
```

Figure 5: GRL model export and import flow

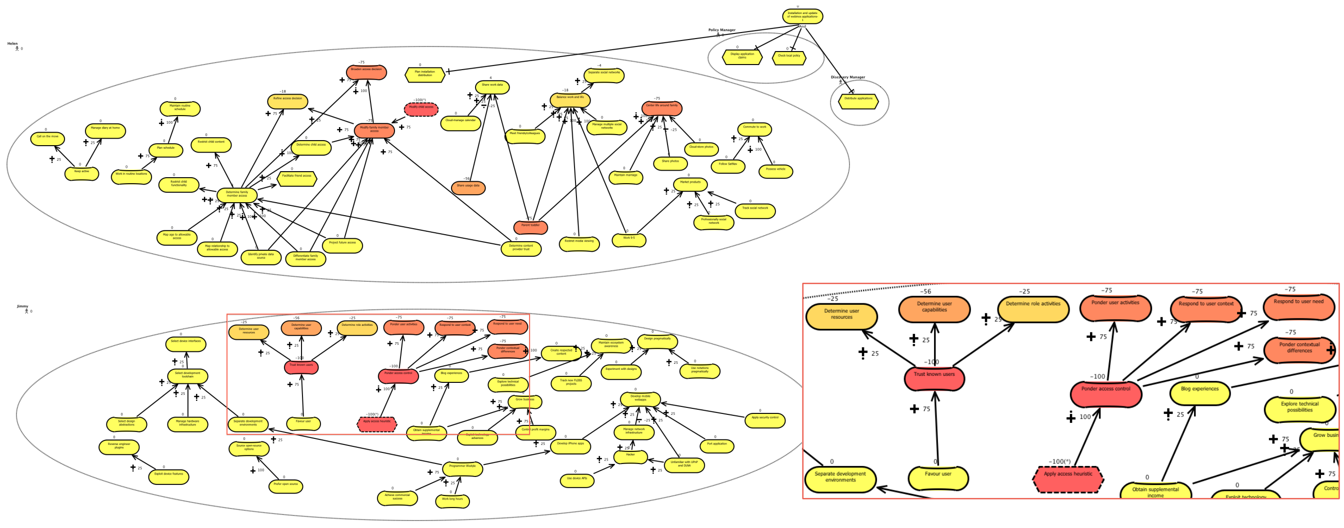


Figure 6: Initial GRL Model generated from the *Helen* and *Jimmy* personas, and *Installation and update of webinos applications* use case (left), with a zoomed portion of the model illustrating the impact of an applied strategy (right)

Using CAIRIS, an initial GRL model was generated to visualise Helen's expectations when installing a webinos application, and Jimmy's expectations when developing the installation procedures for the same application. As Figure 5 shows for the highlighted element in Figure 4, the information captured by CAIRIS about intentions and contribution links is used to generate GRL which, when imported into jUCMNav, is interpreted as a visual model.

The complete version of this initial model is shown in Figure 6 (left). Although the predominant actors in the model are human (Helen and Jimmy), two software based actors (Policy Manager and Discovery Manager) and associated tasks are also created; these indicate the responsibility webinos software components also have in the installation process. Further discussion on the role and nature of these latter actors, and how GRL is derived from the use case is beyond the scope of this paper.

To begin interpreting this model, we applied a strategy to explore the implications of Jimmy having difficulty making decisions about the data 'Kids in Focus' users need access to (dissatisfaction of the *Apply access heuristic* task), and becoming doubtful about the confidence he has about his users' expectations (dissatisfaction of the *Trust known users* softgoal). The strategy also explores the implications of Helen finding difficulty applying access control for 'Kids in Focus' (dissatisfaction of the *Modify child access* and *Share usage data* goals), and the cognitive difficulty Helen might have simultaneously thinking about access control while thinking about her young son (minor dissatisfaction of the *Share usage data* and *Parent toddler* softgoals).

On viewing the initial model, we identified an additional contribution link between goals associated with Helen as well as Jimmy. Helen held a *Determine content provider trust* goal to indicate her belief that content providers have full control over all aspects of access control. This goal appeared to positively contribute to Helen's *Share usage data* goal; this goal was associated with a pre-existing characteristic indicating that Helen would be prepared to share usage information if this led to a positive user experience.

To explore the trust dependencies between Helen and Jimmy, we developed the premortem scenario below. This drew not only on Helen and 'Kids in Focus' but also the software installation process that Jimmy was responsible for maintaining.

Helen installed 'Kids in Focus' on her laptop so Eric could play with Peter (Helen's father) while travelling. A month after installing the application, Helen went on a road trip to a friend. Helen drove into a service station to get some sweets for Eric, but left Eric in the car. When he returned, Eric was gone. Police found that Eric had been playing with a new friend on 'Kids in Focus' in the days leading up to the trip. Forensic investigators believe that some aspect of the application installation process allowed a stranger to obtain private information about Helen and her child. How might this have happened?

We presented this premortem to a webinos team member familiar with 'Kids in Focus', who proposed the following causes.

- Helen over-shared access to her devices with whole world, and the 'Kids in Focus' service was randomly found.
- Helen's account was hijacked and the kidnapper added himself as a valid contact.
- Helen's father inadvertently sent the application to the kidnapper, who captured analytics about Helen; he knew enough to dis-aggregate useful information from the analytics.
- An attacker found a vulnerability in the open source application.

On considering the 3rd cause, we identified missing, but plausible, contribution and dependency links that facilitated this premortem. These included a contribution link between

Jimmy's *Ponder access control* goal, to his *Trust known users* soft goal; this indicates that Jimmy's pondering on the complexity of access control places unwarranted trust on his end-users. We also identified that Jimmy's *Determine user capabilities* goals depends on a goal that Helen has to *Share usage data*, where the dependum is analytics data. Up until this point, the nature of the analytics data, and the unwarranted trust that Helen might have in what Jimmy needs to collect had not been considered.

5. DISCUSSION

Although illustrative, the case study example in Section 4 is comparatively trivial; it considers only two personas where differences in expectations can be identified from inspecting the trust characteristics alone. The framework is, however, scalable to activities involving more than two personas. This is particularly useful in critical systems where some level of assurance is needed about the trustworthiness of personas to undertake certain activities. However, additional techniques may be necessary for making sense of very large GRL models.

An unintentional benefit of this approach is that it encourages designers to use and engage in personas in more depth than they might using personas as a communication tool alone. This is important because, even when used in conjunction with other artefacts like scenarios and cognitive walkthroughs, personas are insufficient for ensuring end-user needs are incorporated into the design process [17]. By forcing designers to use personas as an analytical tool, the success of this approach relies on personas being actively, rather than passively, adopted.

This approach is of particular benefit to social software engineers that wish to explore and visualise the relationship between different aspects of socialness and the software they develop. Although this approach focused on trust expectations, designers may wish to develop social characteristics that appeal to a user populations incentives for engagement, social awareness, or other as yet unforeseen scenarios in the social software engineering research agenda. In the same way that Riegelsberger's framework for trust in technology mediated interactions provided a basis for guiding the analysis necessary to build trust characteristics, other social informatics theories form the basis for supporting a grounded theory analysis for creating and visualising different characteristics of interest.

6. CONCLUSION

This paper presented an approach for eliciting and visualising differences between trust expectations using persona trust characteristics, goal models, and complementary tool support. In doing so, we have made three contributions.

First, we have illustrated how Riegelsberger's framework for trust in technology-mediated interactions helps derive trust characteristics for personas. In doing so, we have shown how our approach can attend to trust at the outset of a project. Moreover, by describing the different facets and expectations of trust using personas, the implications of trust can be made transparent to different stakeholders. Although the qualitative data analysis used in this approach can be time-consuming, it is little more time consuming than creating personas using the Persona Case framework.

Second, we have demonstrated that analysable goal models can be derived from these characteristics. These models further increase the transparency of trust by visualising how goal contributions and dependencies influence a persona's propensity to trust, and how warranted his/her assumptions might be given the characteristics of other personas.

Finally, we have described a case study to illustrate the validity of our approach. We accept that this approach is not a panacea for eliciting all possible trust implications for a given system. Moreover, for social software engineers to adopt this approach, they will need to add personas and i* based modelling languages to their repertoire of design techniques. While the approach makes the creation of personas no less time consuming, we have shown how, with the aid of software tools, this effort can be leveraged to automatically generate large goal models that might otherwise be slow and cumbersome to create.

Acknowledgment

The research was funded by the EPSRC *EUSTACE* project (R24401/GA001), and the EU FP7 *webinos* project (FP7-ICT-2009-05 Objective 1.2).

7. REFERENCES

[1] D. Amyot, S. Ghanavati, J. Horkoff, G. Mussbacher, L. Peyton, and E. Yu. Evaluating goal models within the goal-oriented requirement language. *International Journal of Intelligent Systems*, 25(8):841–877, 2010.

[2] A. K. Chopra. Social computing: Principles, platforms, and applications. In *Requirements Engineering for Social Computing (RESC), 2011 First International Workshop on*, pages 26–29, 2011.

[3] J. Cleland-Huang. Meet elaine: A persona-driven approach to exploring architecturally significant requirements. *IEEE Software*, 30(4):18–21, 2013.

[4] A. Cooper. *The Inmates Are Running the Asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity (2nd Edition)*. Pearson Higher Education, 1999.

[5] J. M. Corbin and A. L. Strauss. *Basics of qualitative research : techniques and procedures for developing grounded theory*. Sage Publications, Inc., 3rd edition, 2008.

[6] W. K. Edwards, M. W. Newman, and E. S. Poole. The infrastructure problem in hci. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 423–432. ACM, 2010.

[7] G. Elahi and E. Yu. Trust trade-off analysis for security requirements engineering. In *Proceedings of the 17th IEEE International Requirements Engineering Conference*, pages 243 –248. IEEE Computer Society, 2009.

[8] S. Faily. Bridging User-Centered Design and Requirements Engineering with GRL and Persona Cases. In *Proceedings of the 5th International i* Workshop*, pages 114–119. CEUR Workshop Proceedings, 2011.

[9] S. Faily. CAIRIS web site. `http://github.com/failys/CAIRIS`, March 2013.

[10] S. Faily and I. Fléchais. Barry is not the weakest link: eliciting secure system requirements with personas. In

Proceedings of the 24th BCS Interaction Specialist Group Conference, BCS '10, pages 124–132. British Computer Society, 2010.

[11] S. Faily and I. Fléchais. Towards tool-support for Usable Secure Requirements Engineering with CAIRIS. *International Journal of Secure Software Engineering*, 1(3):56–70, July-September 2010.

[12] S. Faily and I. Fléchais. Persona cases: a technique for grounding personas. In *Proceedings of the 29th international conference on Human factors in computing systems*, pages 2267–2270. ACM, 2011.

[13] S. Faily and I. Fléchais. User-centered information security policy development in a post-stuxnet world. In *Proceedings of the 6th International Conference on Availability, Reliability and Security*, pages 716–721. IEEE Computer Society, 2011.

[14] S. Faily and J. Lyle. Guidelines for integrating personas into software engineering tools. In *Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems*, EICS '13, pages 69–74. ACM, 2013.

[15] S. Faily, J. Lyle, and S. Parkin. Secure system? challenge accepted: Finding and resolving security failures using security premortems. In *Designing Interactive Secure Systems: Workshop at British HCI 2012*, 2012.

[16] I. Fléchais, J. Riegelsberger, and M. A. Sasse. Divide and conquer: the role of trust and assurance in the design of secure socio-technical systems. In *NSPW '05: Proceedings of the 2005 workshop on New security paradigms*, pages 33–41. ACM, 2005.

[17] E. Friess. Personas and decision making in the design process: an ethnographic case study. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, pages 1209–1218. ACM, 2012.

[18] C. Fuhrhop, J. Lyle, and S. Faily. The webinos project. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 259–262. ACM, 2012.

[19] P. Giorgini, H. Mouratidis, and N. Zannone. Modelling Security and Trust with Secure Tropos. In H. Mouratidis and P. Giorgini, editors, *Integrating Security and Software Engineering*. Idea Group, 2007.

[20] N. Maiden, S. Jones, C. Ncube, and J. Lockerbie. Using i* in Requirements Projects: Some Experiences and Lessons. In E. Yu, editor, *Social Modeling for Requirements Engineering*. MIT Press, 2011.

[21] D. L. Moody, P. Heymans, and R. Matulevicius. Improving the effectiveness of visual representations in requirements engineering: An evaluation of i* visual syntax. In *Proceedings of the 17th IEEE International Requirements Engineering Conference*, pages 171–180. IEEE Computer Society, 2009.

[22] G. Mussbacher, S. Ghanavati, and D. Amyot. Modeling and Analysis of URN Goals and Scenarios with jUCMNav. In *Proceedings of the 2009 17th IEEE International Requirements Engineering Conference, RE*, RE '09, pages 383–384. IEEE Computer Society, 2009.

[23] E. Paja, F. Dalpiaz, P. Giorgini, S. Paul, and P. H. Meland. Modelling Trust and Security Requirements: the Air Traffic Management Experience. In *Proceedings of iStar Showcase*, 2011.

[24] J. Riegelsberger, M. A. Sasse, and J. D. McCarthy. The mechanics of trust: A framework for research and design. *International Journal of Human Computer Studies*, 62(3):381–422, 2005.

[25] webinos Consortium. Kids in Focus Concept Application. `https://github.com/webinos-apps/app-kids-in-focus`, January 2013.

[26] webinos Consortium. webinos design data repository. `https://github.com/webinos/webinos-design-data`, March 2013.

[27] webinos Consortium. webinos personas. `https://github.com/webinos/webinos-design-data/tree/master/personas`, March 2013.

[28] E. Yu. Towards modeling and reasoning support for early-phase requirements engineering. In *Proceedings of the 3rd IEEE International Symposium on Requirements Engineering*, pages 226–235. IEEE Computer Society, 1997.

[29] D. E. Zand. Trust and managerial problem solving. *Administrative Science Quarterly*, 17(2):229–239, 1972.