Visualising personas as goal models to find security tensions

Shamal Faily
Department of Computing and Informatics, Bournemouth University, Poole, UK
Claudia Iacob
School of Computing, University of Portsmouth, Portsmouth, UK
Raian Ali
Hamad Bin Khalifa University, Doha, Qatar, and
Duncan Ki-Aries
Department of Computing and Informatics, Bournemouth University, Poole, UK

Received 12 March 2021
Revised 25 March 2021
Accepted 26 March 2021

Abstract

Purpose – This paper aims to present a tool-supported approach for visualising personas as social goal models, which can subsequently be used to identify security tensions.

Design/methodology/approach – The authors devised an approach to partially automate the construction of social goal models from personas. The authors provide two examples of how this approach can identify previously hidden implicit vulnerabilities and validate ethical hazards faced by penetration testers and their safeguards.

Findings – Visualising personas as goal models makes it easier for stakeholders to see implications of their goals being satisfied or denied and designers to incorporate the creation and analysis of such models into the broader requirements engineering (RE) tool-chain.

Originality/value – The approach can be used with minimal changes to existing user experience and goal modelling approaches and security RE tools.

Keywords Personas, Goal models, i*

Paper type Research paper

1. Introduction

Software products and services cannot be secure unless they are usable (Association for Computer Machinery, 2018), yet too often security and usability are considered as a trade-off, i.e. you cannot have one without sacrificing the other. In reality, security is a human value situated in a delicate balance with other values, such as privacy and trust; touching one value implicates others (Friedman and Hendry, 2019). User research is necessary to elicit these tensions, but – given their impact – we must capture how these tensions impact some broader theory or system of concern.

Information & Computer Security
Vol. 29 No. 5, 2021
pp. 787-815
Emerald Publishing Limited
2056-4961
DOI 10.1108/ICS-03-2021-0035

Personas are fictional characters that represent archetypal users, and embody their needs and goals (Cooper *et al.*, 2014). Personas are the product of research with representative end-users; designing for a single persona means designing for the user community he or she represents. By facilitating design for one customer voice rather than many, personas have become a popular user experience (UX) technique for eliciting and validating user requirements. They are also helpful when designing for security. Should a persona experience physical or cognitive burden whilst completing a task then his performance might not be as intended. Steps might be omitted or the task altered to achieve an end more conducive to the persona's own goals, irrespective of whether or not the intent is malicious.

Personas can inspire the identification of security tensions, but in practice, they do not. Design processes prioritising agility provide little time for using personas for anything besides validating stakeholder value has been achieved. Even if we assume UX and security engineers collaborate, personas are not always used in the ways envisaged by designers (Friess, 2012), whilst security engineers might primarily focus on requirements for security mechanisms. Given their differing concerns and perspectives, problems may not be found even when these are indicated during the collection or analysis of user research data.

Personas, as user models, can be integrated into security requirements engineering (RE) practices and tools, but they need to be built and presented differently. This may make it easier for stakeholders to identify the security implications of user goals being satisfied or denied. Goal models in languages such as *i** (Yu, 1997) and the goal-oriented requirements language (GRL) (Amyot *et al.*, 2010) provide a foundation for this improved integration; they represent the intentions and rationale of social and technical actors, their inter-relations and alternative strategies giving space for variability accommodation, including that of user types. Approaches such as Secure Tropos (Mouratidis and Giorgini, 2007) and socio-technical systems-ml (Paja *et al.*, 2013) show how goal models can be used in the early stage of design to find vulnerabilities. However, they are role-focussed whereas people are expected to align to one or more ways to achieve predefined goals.

To integrate personas into goal-oriented security RE, this paper presents a tool-supported approach for visualising personas as goal models, and extends previous work presented at the fourth International Workshop on Security and Privacy RE (Faily *et al.*, 2020a). In doing so, we provide two examples of how this approach can identify security tensions. User research and threat modelling can be time-consuming and cognitively intensive activities that might happen separately or in parallel before, during or after other RE activities. It is, therefore, necessary to loosely couple these goal models such that other design models can evolve orthogonally with minimum disruption to existing processes and tools.

The remainder of this paper is structured as follows. In Section 2, we consider related work in social goal modelling and security, personas and usable and secure RE upon which our approach is based. In Section 3, we present the processes and tool-support algorithm that underpins our approach. Using an industrial control systems case study example, we provide two examples of how this approach can be used to identify security tensions in Section 4. We discuss the implications of our work and potential limitations in Section 5, before concluding in Section 6 by summarising the contributions of our work to date and directions for future work.

2. Related work

2.1 Personas for security

UX professionals have long used personas to bring user requirements to life, and there has been some work within the RE community on using personas to add contextual variability to social goal models (Nunes Rodrigues *et al.*, 2018).

The merits of using personas to elicit security requirements were identified by Faily and Fléchais (2010), who showed how the use of personas could show the human impact of security to stakeholders who have never met user communities represented by personas. In recent years, there has also been additional interest in the RE community on the use of personas to engage stakeholders when validating requirements (Cleland-Huang, 2013), and how data used to construct personas can have some security value. For example, Mead *et al.* (2017) demonstrated how the text from personas built on assumptions about attackers – Personae Non-Gratae – could be mined to identify potential threat models and identify gaps between a designer's and attacker's model of a system. However, Mead and her colleagues focus on the identification of threats to a system rather than vulnerabilities that might arise from interactions between personas and the system.

2.2 Assuring personas with argumentation models

To use personas in some security context, we need confidence they are valid. Without this assurance, the legitimacy of any models related to them could be called into question. Chapman and Milham (2006) question whether obtaining this validity is possible. They claim applying qualitative approaches to the creation of personas means personas cannot be validated because the personas cannot be falsified and subsequently disproved. However, subsequent work in the human computer interaction (HCI) community found that, if personas are constructed using qualitative data analysis, the results of this analysis can be framed as argumentation models (Faily and Fléchais, 2011). These models, based on Toulmin's model of argumentation (Toulmin, 2003), draw analogies between the characteristics of a persona and a *claim* underpinned by multiple propositions as *grounds*, *warrants* that connect these grounds to a claim and *rebuttals* that state propositions questioning the basis of a claim. Argumentation models have also made an important contribution to evaluating the rationale underpinning software design, and providing confidence that claims made during software design can be justified (Burge *et al.*, 2008).

2.3 Finding vulnerabilities using social goal modelling

Personas are not the only models used for modelling users and their expectations. Social goal modelling languages such as *i** have been proposed by the RE community for specifying human intentions, and modelling how they contribute to other intentions. They also support the modelling of dependencies between different actors, where a *depender* actor depends on *dependee* actor for some resource *dependum*. Such dependencies allude to broader vulnerabilities because actors themselves become vulnerable when they rely on dependees for dependums. Analysing the chains of these dependencies can indicate how vulnerable these actors are (Yu, 1995). Moreover, when such models capture a socio-technical system of actors and resources, they also highlight potential system vulnerabilities resulting from inconsistencies between an organisation's policies and working practices (Massacci and Zannone, 2011).

In previous work examining the use of social goal modelling to support security RE, Liu *et al.* (2003) considered how legitimate actors might use their intentions, capabilities and social relationships to attack the system and how dependency relationships form the basis of exploitable vulnerabilities. The idea of dependencies as implicit vulnerabilities was further elaborated by Giorgini *et al.* (2005), who indicated that dependency relationships can also capture trust relationships where dependers believe dependees will not misuse a goal, task or resource (trust of permission) or a trustee believes dependees will achieve a goal, execute a task or deliver a resource (trust of execution).

Elahi *et al.* (2010) incorporated vulnerabilities into goal models to link knowledge about threats, vulnerabilities and countermeasures to stakeholder goals and security requirements.

Vulnerabilities are considered as weaknesses in the structure of goals and activities of intentional agents, which can be propagated via decomposition and dependency links. The introduction of vulnerabilities was added on the basis that including security and non-security elements on a single model makes models clearer and facilitates model discussion (Sindre and Opdahl, 2007). However, whilst this approach supports the *specification* of vulnerabilities, it provides little support for *eliciting* them. This still requires a priori knowledge of potential system weaknesses or threat models that could take advantage of them. Moreover, Moody *et al.* (2009) found that the graphical complexity of *i** is several times greater than a human's standard limit for distinguishing alternatives. As such, approaches that increase the complexity of the *i** language are likely to hinder rather than improve the understandability of social goal models, particularly for novices.

2.4 Integrating requirements and information security and computer-aided integration of requirements and information security

The need to elicit, specify and validate requirements for usable and secure systems has been independently visualised by the RE, security and HCI communities. As an exemplar for showing how concepts from these areas complement each other, IRIS (integrating requirements and information security) framework (Faily, 2018) was devised. IRIS is a process framework for designing usable and secure software. It incorporates a methodology agnostic meta-model for usable and secure RE that supports the complementary use of different security, usability and RE techniques. Personas are integrated into this framework, which uses the knowledge acquisition in autOmated specification (KAOS) language for modelling system goals (van Lamsweerde, 2009), obstacles that obstruct the satisfaction of these goals, dependency associations between roles and relationships between tasks, system goals and the roles responsible for them. The framework is complemented by CAIRIS (computer-aided integration of requirements and information security): a software platform for eliciting, specifying, automatically visualised and validating secure and usable systems that is built on the IRIS meta-model. By making explicit the links between different security, usability and software models using IRIS and providing tool-support for automating generating and validating these models, IRIS and CAIRIS can put one model in context with another. For example, we recently demonstrated how data flow *taint* can be identified in data flow diagrams within CAIRIS by putting these diagrams in context with other software and usability models (Faily *et al.*, 2020b).

Previous work has shown that persona characteristics can be re-framed as goals and soft goals in social goal models (Faily, 2011). Not only does this make it possible to automatically generate goal models from argumentation models but also some assurance is also provided for both the basis of user goals and the broader impact of satisfying these goals on other system elements. Subsequent work has demonstrated how these concepts can lead to the generation of elaborate GRL compatible goal models (Faily and Fléchais, 2014). However, a weakness of this approach is its reliance on additional tool support (jUCMNav), and the limited support of traceability links between the goal modelling platform and its originating data should the GRL model evolve; such evolution is likely as different stakeholders make sense of this model. Subsequent refinement of the jUCMNav model could lead to additional effort by analysts to ensure the goal model and its foundational CAIRIS models are visualised.

3. Approach

3.1 Conceptual model

To reframe a persona as a social goal model, our approach relies on aligning concepts from IRIS with concepts from social goal modelling. A review of the complete conceptual model, which is described in more detail in Faily (2018), is beyond the scope of this paper. We do,

however, summarise this model concept alignment in Figure 1, which we provide further rationale for in the sub-sections below.

3.1.1 Personas and persona characteristics. Our approach has only a minimal impact on existing IRIS concepts. We assume personas consist of multiple persona characteristics. These characteristics are attributes of persona behaviour; they can be considered as arguments for persona behaviour, and are grounded in one or more grounds, warrants and rebuttal elements. These elements (document references) are *factoids* that can be drawn from a variety of primary and secondary data sources (external documents) such as interview transcripts, observational notes and websites. Further details on these concepts can be found in Faily (2018).

3.1.2 User goals. User goals represent the intentional desires of actors, where actors are personas. This definition is in line with the definition used for goals by the social goal modelling community (Yu, 1995). In our approach, user goals are factoids expressed intentionally. Yu *et al.* (2011) state that intentional properties can only be inferred based on information obtained by indirect means, and that the validity of these attributions can never be certain. However, as indicated in Section 2.2, the qualitative underpinnings of personas *can* be validated, in the same way, that qualitative models, in general, can be validated. So, although validity can never be certain, our model provides some level of assurance. Based on the satisfaction levels proposed by Amyot *et al.* (2010), user goals can be assigned a qualitative satisfaction level associated with a quantitative score; these values are Satisfied (100), Weakly Satisfied (50), Weakly Denied (−50) and Denied (−100).

3.1.3 Hard/soft goals and beliefs. Our approach inherits the idea of hard goals, soft goals and beliefs from *i**.

Hard goals are goals that can be measurably satisfied, whereas soft goals are goals with less well-defined success criteria that can be satisfied (Simon, 1979).

Beliefs capture facts important to stakeholders (Amyot *et al.*, 2010); we use these to capture beliefs held by personas. Beliefs are used irregularly in goal models, and whilst it has been suggested these are used to capture the rationale of designers during modelling

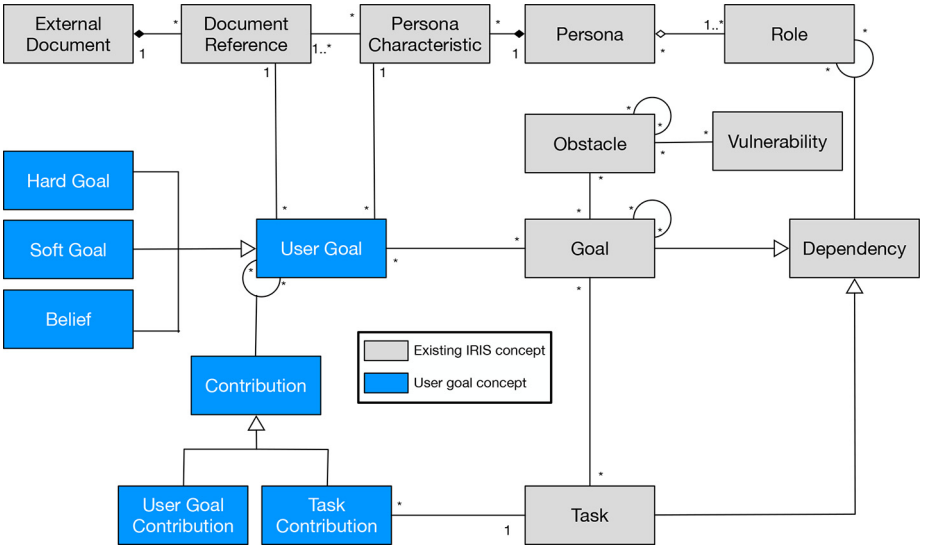


Figure 1.
UML class diagram of IRIS and user goal concepts

rather than stakeholders (Regev and Wegmann, 2005), it has also been accepted that further exploration on the semantics of beliefs is needed (Yu, 2009). The grounding of personas and IRIS' support for KAOS domain properties – that can capture this form of rationale – means we need not explicitly incorporate rationale meta-data into visual models. Therefore, beliefs can safely be used to represent stakeholder beliefs without confusion. User goals are elicited from persona characteristic elements based on the trust characteristic elicitation process described in Faily and Fléchais (2014), where the implied goal, soft goal or belief intentions form the basis of user goals associated with the characteristic and its grounds, warrants and rebuttal elements. These user goals are expressed as persona intentions.

3.1.4 Aligning system and user goals. As Figure 1 shows, IRIS supports the concept of system goal, i.e. prescriptive statements of intent that the system should satisfy throughout the cooperation of its intended roles in a particular environment; this definition is based on the KAOS definition of goal (van Lamsweerde, 2009). Obstacles obstructing these goals may be associated with vulnerabilities, thereby connecting a goal view of a system with a risk view. IRIS also supports dependency modelling of system goals, where a *depender* role depends on *dependee* role for a goal or task *dependum*.

Until now, IRIS has not incorporated the notion of user goal because, as a methodologically agnostic meta-model, discretion on how to map user goals and expectations to system functionality is left to designers. However, in the case of a goal dependum, we should be able to capture the need for user goals to be satisfied to satisfy system goals. Consequently, our approach now adds an explicit traceability link between user goals that personas might have, and KAOS goals that a system needs to satisfy. This traceability link could be bi-directional, as we do not prescribe the elicitation of one type of goal before the other. For example, an analyst may capture system goals to satisfy a persona's goals, so may wish to indicate the system goals that address these user goals. Conversely, in a pre-existing system model, an analyst may wish to examine the implication of system requirements on the value a persona wishes to achieve. Our approach precludes neither possibility and facilitates subsequent model validation checks.

3.2 Modelling user goal contributions

To visualise personas as goal models, our approach extends the *i** Strategic Rationale model (Yu, 1995) in two ways.

Firstly, we align persona characteristic elements with *contribution* links. Contribution links indicate the desired impact that one system element has on another (Amyot *et al.*, 2010). As user goals are part of the broader socio-technical system being modelled, it is reasonable to assume that one user goal can contribute to another. In our approach, argumentation elements form the basis of means/end contribution links between user goals, i.e. where one user goal is the *means* for another user goal's *end*. Links are annotated with two additional pieces of information, namely, whether a link is a "means" or an "end" with respect to the characteristic's goal, soft goal or belief, an optional initial satisfaction level, based on the qualitative values and quantitative scores specified in Amyot *et al.* (2010), i.e. Satisfied (100) Weakly Satisfied (50), Weakly Denied (−50) and Denied (−100); this is analogous to the setting of *strategies* in jUCMNav (Amyot *et al.*, 2010).

Secondly, as tasks can have a security impact (Elahi *et al.*, 2010), completion of a task contributes to one or more user goals.

```
   Input  : goalName - the goal name
   Data: evaluatedGoals - set of previously evaluated goals and their contribution scores, cts - names of tasks contributing
         to user goal goalName, cgs - names of user goals contributing to user goal goalName, linkScore - quantitative
         score for the contribution of user goal cgName to user goal goalName, contScore - product of linkScore and the
         goal contribution score for user goal cgName
   Output: score - contribution score
 1 Function calculateGoalContribution(goalName) is
 2     score ← initialSatisfactionScore goalName;
 3     if score = 0 then
 4         isObstructed ← systemGoalObstructed goalName;
 5         if isObstructed then
 6             score ← -100;
 7         else
 8             if goalName ∉ domain evaluatedGoals then
 9                 cts ← taskLinks  goalName;
10                 while taskName ← cts do
11                     score ← score + taskContributionScore taskName;
12                 end
13                 cgs ← goalContributions goalName;
14                 while cgName ← cgs do
15                     linkScore ← contributionLinkScore goalName cgName;
16                     cgScore ← calculateGoalContribution cgName;
17                     contScore ← linkScore × cgScore ;
18                     score ← score + contScore;
19                 end
20                 score ← score / 100;
21                 if score < -100 then
22                     score ← -100;
23                 else if score > 100 then
24                     score ← 100;
25                 end
26                 evaluatedGoals ← evaluatedGoals ∪ {goalName → score};
27             else
28                 score ← evaluatedGoals goalName;
29             end
30         end
31     end
32     return score;
33 end
```

Like other goal modelling languages, contributions have a qualitative value corresponding to a quantitative score. We base these values on those used by GRL: Make (100), SomePositive (50), Help (25), Hurts (−25), SomeNegative (−50) and Break (−100). Make and break contributions lead to the satisfaction or denial of user goals, respectively; similarly, help and hurt contributions help or hinder the satisfaction of user goals. SomePositive and SomeNegative values indicate some indeterminate level of positive or negative contribution that exceeds helping or hindering.

The approach for calculating contributions is similar to Giorgini *et al.*'s label propagation algorithm (Giorgini *et al.*, 2003). We implemented a recursive, forward propagation *calculateGoalContribution* (Algorithm 1) based on the *CalculateContribution* algorithm described in Amyot *et al.* (2010).

The setting of an initial satisfaction score (Line 2) based on the previously described satisfaction level is permitted; this can override the calculated goal score from related tasks and goal contributions. If the initial satisfaction score has not been overridden and no system goals associated with a user goal have not been obstructed (Lines 4–6), a contribution score is calculated. To handle goal contribution loops, i.e. where user goal x is a means to goal y, which is a means to goal x or situations where the user goal x contributes to several user goals that eventually contribute to user goal y, a persistent set of visited goals and their contribution scores, *evaluatedGoals*, is retained. Propagation occurs if a goal's name is not in this set (Lines 9–26), otherwise

the previously retained contribution for that goal is reused (Line 28). The contribution score is calculated based on the tasks contributing to it (Lines 9–12) and the product of each contributing goal and the contribution link strength (Lines 13–19). If the score calculated is greater than 100 or less than −100 then the score is visualised to a value within this range (Lines 21–25).

3.3 Tool-support

To show how this approach might be implemented in requirements management tools more generally, we incorporated a new model type and supporting tools into CAIRIS release 2.3.6.

We tool-supported the additional concepts and algorithms by introducing a *user goal* visual model. This is based on the visual semantics of GRL, where a rounded box represents a hard goal, a polygon with rounded corners represents a soft goal, an ellipse represents a belief and dashed rectangle models the actor boundary. In this model, actors are represented by personas. Further drawing from the semantics used by GRL and jUCMNav, these nodes are coloured from dark green to dark red corresponding with satisfaction values of Satisfied (100) and Denied (−100); nodes with a value of None (0) are coloured yellow.

User goal models are generated automatically by CAIRIS using the same pipeline process used to visualise other CAIRIS models. A declarative model of graph edges is generated by CAIRIS; this is processed and annotated by Graphviz (AT&T, 2020) before being subsequently rendered as scalable vector graphics. This annotation stage includes applying Algorithm 1 to user goal nodes to determine its score, and subsequent colour. The CAIRIS model generation process is described in more detail in Faily (2018). The algorithms described were incorporated into an *implied vulnerability* model validation check, which is applied to all KAOS goal dependency relationships in a CAIRIS model. CAIRIS model validation checks are implemented internally within the relational database used by a CAIRIS model as structured query language stored procedures.

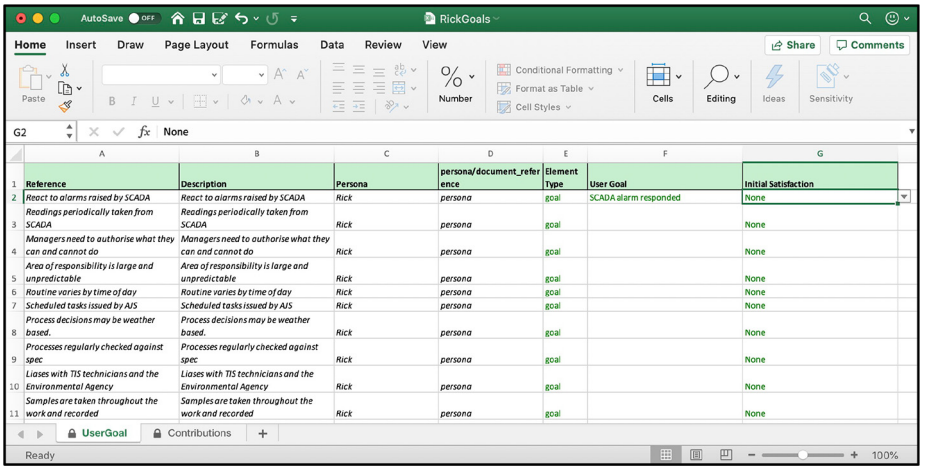
As shown in Figure 2, we also extended CAIRIS to generate Excel workbooks for capturing user goals and contribution links. Such workbooks are useful for analysts wishing to contribute to user goal modelling via more familiar office automation tools.

The generated Excel workbook contains UserGoal and UserContribution spreadsheets, where edited cells for both are coloured green. The UserGoal worksheet is pre-populated with read-only data on the persona characteristic or document reference name, its description, the persona it is associated with and an indicator to whether the reference corresponds to a persona [characteristic] or document reference. When completing the worksheet, analysts should indicate the intentional elements associated with the persona characteristics or document references providing their grounds, warrants or rebuttals. Analysts should also indicate the element type (goal, softgoal or belief), and the initial satisfaction level using the dropdown lists provided. The source and destination cells in the ContributionsSheet are pre-populated once user goals have been added in the UserGoal sheet, so only the means/end and contribution links need to be set.

We further extended CAIRIS to allow the contents of these workbooks to be imported into a pre-existing CAIRIS model.

4. Case study: security tensions at ACME water

We show how our approach might be used with two examples[1] that identify security tensions associated with *ACME Water*. *ACME Water* is an visualised UK water company



Reference	Description	Persona	persona/document_reference	Element Type	User Goal	Initial Satisfaction
React to alarms raised by SCADA	React to alarms raised by SCADA	Rick	persona	goal	SCADA alarm responded	None
Readings periodically taken from SCADA	Readings periodically taken from SCADA	Rick	persona	goal		None
Managers need to authorise what they can and cannot do	Managers need to authorise what they can and cannot do	Rick	persona	goal		None
Area of responsibility is large and unpredictable	Area of responsibility is large and unpredictable	Rick	persona	goal		None
Routine varies by time of day	Routine varies by time of day	Rick	persona	goal		None
Scheduled tasks issued by AJS	Scheduled tasks issued by AJS	Rick	persona	goal		None
Process decisions may be weather based.	Process decisions may be weather based.	Rick	persona	goal		None
Processes regularly checked against spec	Processes regularly checked against spec	Rick	persona	goal		None
Liases with TIS technicians and the Environmental Agency	Liases with TIS technicians and the Environmental Agency	Rick	persona	goal		None
Samples are taken throughout the work and recorded	Samples are taken throughout the work and recorded	Rick	persona	goal		None

Figure 2.
Generated Excel workbook for entering user goals and contributions

responsible for providing clean and wastewater services to several million people in a particular UK region. The infrastructure needed to support such a large customer base was substantial, amounting to over 60 water treatment works, 800 wastewater treatment works, 550 service reservoirs, 27,000 km of water mains, 19,000 km of sewer networks, with over 1,900 pumping stations and 3,200 combined sewer outflows.

The first example in Section 4.1 describes and illustrates how our approach complements the steps necessary to identify implicit vulnerabilities affecting a security policy. *Implicit* vulnerabilities are vulnerabilities that may be present when dependees fall short of their responsibility to deliver dependums (Liu *et al.*, 2003). Where these vulnerabilities are present, the values held by humans acting as dependers and dependees may be in tension with trust. Consequently, identifying these tensions may provide insights into the root causes of system vulnerabilities.

The second example in Section 4.2 shows how our approach can be used to examine security tensions associated with *penetration testing*. Penetration testers conduct visualised "penetration tests" that probe a system's defences to evaluate the impact of any discovered weaknesses; they require creativity and ingenuity to find unexpected ways of breaching a system, with the added constraint that finding and exploiting vulnerabilities should neither harm the system nor encroach on the dignity of those affected by it. When faced with ethical dilemmas, penetration testers are expected to adopt different ethical perspectives when deciding the right course of action (Mouton *et al.*, 2015). However, previous work by some of the authors (Faily *et al.*, 2015) indicated that penetration testers are subject to certain decision-making biases; when faced with ethical dilemmas, these biases influence decisions of ethical import.

4.1 Example 1: implicit vulnerabilities affecting ACME water security policy

4.1.1 Implicit vulnerability identification. The steps are taken to identify implicit vulnerabilities are concerned with dependencies between systems rather than user goals, and considers two situations where dependums might not be delivered. Firstly, if a system goal dependum or its refinements are obstructed and not resolved. Secondly, if the dependum or its refinements are linked with denied user goals.

```
   Input  : g - the goal name
   Data: ugs - names of user goals linked to system goal g, goals - names of system goals refinements of g, obs - names of
         obstacles obstructing system goal g
   Output: isObstructed - indicates if goal g is obstructed
 1 Function isGoalObstructed(g) is
 2 |   isObstructed ← false;
 3 |   ugs ← linkedUserGoals g;
 4 |   while ug ← ugs do
 5 |   |   score ← calculateGoalContribution ug [];
 6 |   |   if score < 0 then
 7 |   |   |   isObstructed ← true;
 8 |   |   |   break;
 9 |   |   end
10 |   end
11 |   if isObstructed = false then
12 |   |   goals ← refinedGoals g;
13 |   |   if goals = ∅ then
14 |   |   |   obs ← obstructingGoals g;
15 |   |   |   if obs ≠ ∅ then
16 |   |   |   |   isObstructed ← true;
17 |   |   |   else
18 |   |   |   |   while o ← obs do
19 |   |   |   |   |   isObstructed ← isObstacleObstructed o;
20 |   |   |   |   end
21 |   |   |   end
22 |   |   else
23 |   |   |   while g ← goals do
24 |   |   |   |   isObstructed ← isGoalObstructed g;
25 |   |   |   end
26 |   |   end
27 |   end
28 |   return isObstructed;
29 end
```

Algorithm 2 specifies how the presence of such implicit vulnerabilities might be identified within a typical recursive system goal satisfaction algorithm. The algorithm returns a value of true if the system goal *g* is obstructed.

The algorithm navigates the visualising tree-based KAOS goal refinements (Lines 11–27) to determine if there are *obstruct* associations between refined goals and obstacles, and these obstacles have not been resolved, i.e. there are no *resolve* relationships between obstacles and goals, which address them. However, this check can be a shortcut should a linked user goal associated with system goal *g* be denied, i.e. has a score less than 0 (Lines 3–10). Should this check not be a shortcut then the *isObstacleObstructed* algorithm (Line 19) determines whether a goal is obstructed. This algorithm returns a value of true should one or more of the following conditions hold, namely, the obstacle or one of its obstacle refinements are not resolved by a [mitigating] system goal, an obstacle or one of its obstacle refinements are resolved, but the resolved goal has one or more linked user goals, which are denied. The *isObstacleObstructed* algorithm is formally specified in Faily *et al.* (2020b).

Vulnerabilities within IRIS are defined as system weaknesses (Faily, 2018), but an implicit vulnerability may not always be a system weakness. It may indicate some inconsistency between what system roles and humans fulfilling might want and need or – as suggested by Pastor *et al.* (2011) – some level of human fallibility resulting from roles that participate in too many dependencies as a depender. However, implicit vulnerabilities can help make sense of different system models and, in doing so, provide a rationale for vulnerabilities feeding into risk models.

4.1.2 User research and persona creation. Four in-situ interviews were held with six plant operators, supervisory control and data acquisition (SCADA) engineers and plant operation managers at two clean water and two wastewater treatment plants. These interviews were recorded, and the transcripts analysed using grounded theory (Corbin and Strauss, 2008).

The analysis led to a qualitative model of plant operations security perceptions. Using the persona case technique (Faily and Fléchais, 2011), we derived, from the model, a single persona of a water-treatment plant operator, Rick, incorporating 32 persona characteristics and backed up by 82 argumentation elements (grounds, warrants or rebuttals).

4.1.3 ACME water security policy model. The ACME Water security policy was modelled as a KAOS goal model where each system goal represented a policy goal. These policy goals were created by visualise existing documentation about ACME's existing information security policy and agreeing the scope of the policy to be modelled with ACME's information technology (IT) security manager. Existing policy documentation was analysed to elicit and specify a KAOS goal model of 82 policy goals, with a single high level goal (*Secure operating environment*) and, as shown in Figure 3, 11 refined sub-goals representing the different policy areas. These goals and other security and usability elements of the operating environment were specified in a CAIRIS model; these included 2 personas, 11 roles, 21 obstacles, 9 vulnerabilities, 5 tasks and 6 role-goal-role dependencies.

4.1.4 User goal model creation. To generate a user goal model based on Rick, we initially derived 104 user goals and beliefs from both the persona characteristics and argumentation elements and 165 contribution links. The first two authors then reviewed the model to de-duplicate synonymous user goals. For example, a *Site protected* user goal was associated with a *Copper theft* document reference, as the intention implied was that the site needed to be protected from this threat. However, we identified a *Site secured* user goal associated with *Physical and login security* document reference. As a result, we deleted the former user goal, and contribution linked its user goals to *Site secure*. In parallel with the de-duplication of user goals, we also added additional contribution links between user goals based on our understanding of the persona and his intentions, where these contribution links cross-cut persona characteristics. For example, on reviewing the persona characteristics and their underpinning data, we noted that the *Thieves ignore impact* user goal, which was associated with the *thieves do not care about their impact* characteristic, helped foster the belief that *personal safety is a hygiene factor*; this belief was associated with the *personal safety is an infosec hygiene factor* persona characteristic. Following this analysis, the final model resulted in 93 user goals and beliefs and 205 contribution links.

Figure 4 shows the goal model generated by CAIRIS for Rick.

4.1.5 ICT awareness implicit vulnerabilities. From Figure 5, we identified a link between the *InfoSec communications perceived* user goal (annotated as 2) and the *inforrmation and communications technology (ICT) awareness* system goal, which is a refinement of the high-level *Secure Site* system goal.

The *ICT awareness* system goal indicates ICT partners should know how to maintain equipment hosted in the secure areas and, as Figure 5 (inset) shows, this system goal is already obstructed due to exposed and surplus equipment, which should not be present. Unfortunately, as Figure 5 also indicates, the related user goal is also denied. The negative impact affects not only the perception of site security but also the perception the site is run efficiently; this corroborates the obstacles found to be present in the system goal model. To reinforce this, the belief *Thieves steal anything* (annotated as 1) was set to satisfied, which weakly denied *InfoSec communications perceived*, further validating negative perception.

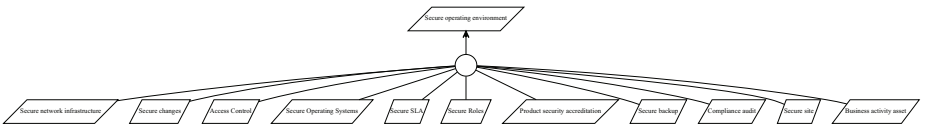


Figure 3.
High-level ACME water security policy goals

This highlighted the need for a new dependency where an IT security manager depends on an ICT partner to achieve the *ICT awareness* goal.

The limited security awareness means operators fail to see the connection between misunderstanding visualised, and wifi insecurity and site security, due to their belief that an air gap exists between wireless networks and industrial control systems. Access controls on pump actions further support the belief that unknown applications are unauthorised. To explore this further, we associated the *Pump action restricted* user goal with the *Access Control* system goal, and added a dependency to indicate that plant operators depend on information security managers for this goal. CAIRIS subsequently flagged a model validation warning because a refined goal *Vendor passwords* was obstructed, due to evidence that vendors were using easily guessed default passwords for certain critical components.

4.1.6 Validating vulnerabilities with implicit vulnerabilities. As indicated in Figure 1, obstacles can be associated with vulnerabilities to capture the rationale for including vulnerabilities in subsequent risk analysis activities. In the ACME water model, an *Exposed ICT Cabinets* obstacle was already associated with an *Exposed cabinet* vulnerability, but – given how divisive resolving obstacles might be because of the architectural implications of their resolution – we wanted to see if the user goal model of Rick provided a human rationale for the obstacle's presence.

Information security managers depend on plant operators for a related *Industrialised secure cabinet* system goal to ensure control systems are kept in secure cabinets. On reviewing the user goal model and the tasks in the ACME water model, we noted that no one was explicitly required to check these cabinets; instead, ACME water trusted Rick to do this whilst discharging other duties.

As Figure 6 shows, as part of a pre-existing *Broken Instrument alarm* task (annotated as 3), we introduced help contribution links to *Complex failure callout* and *SCADA alarm responded* because Rick completes the task to satisfy these user goals. The task entails Rick being away from the safety of the control room to respond to equipment alarms from these cabinets. Should these alarms fire out of hours, the model shows that Rick might feel uneasy, particularly if he thinks the alarm indicates intruders are stealing equipment. The potential for Rick to skip the steps necessary to check these cabinets was corroborated in the user goal model due to the *SCADA alarm responded* being very weakly satisfied.

4.2 Security tensions resulting from penetration testing practices

4.2.1 User research and persona creation. We visualised a penetration tester persona to examine the implications of the interaction between testers and their tools and techniques, typify the situations where such decisions might be made, and identify penetration tester goals that positively or negatively impact these situations.

We analysed transcripts collected during the previous work by the authors (Faily *et al.*, 2015) from eight semi-structured interviews with professional penetration testers; each interview lasted approximately 45 min. Using grounded theory (Corbin and Strauss, 2008), we analysed the transcripts and develop a qualitative model of ethical hazards and safeguards. Based on this model, we elicited the four *ethical hazards* specified in Table 1. These are situations likely to increase the probability of unethical behaviour because of the means, motive and opportunity to engage in such behaviour (Pendse, 2011). These ethical hazards can be mitigated by the safeguards summarised in Table 2.

Using the persona case process (Faily and Fléchais, 2011) and CAIRIS, a persona was created based on an experienced penetration tester, Ben, incorporating 18 persona characteristics and backed up by 84 argumentation elements. The persona was distributed

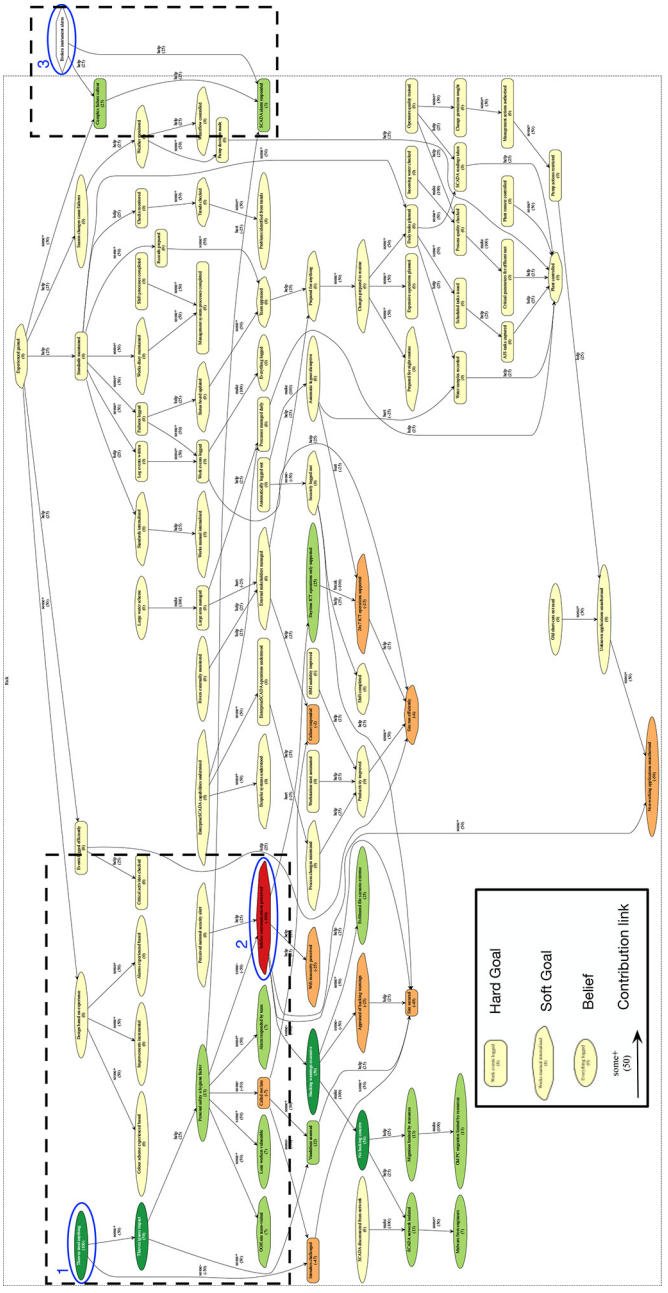


Figure 4.
Annotated CAIRIS user goal model based on rick persona

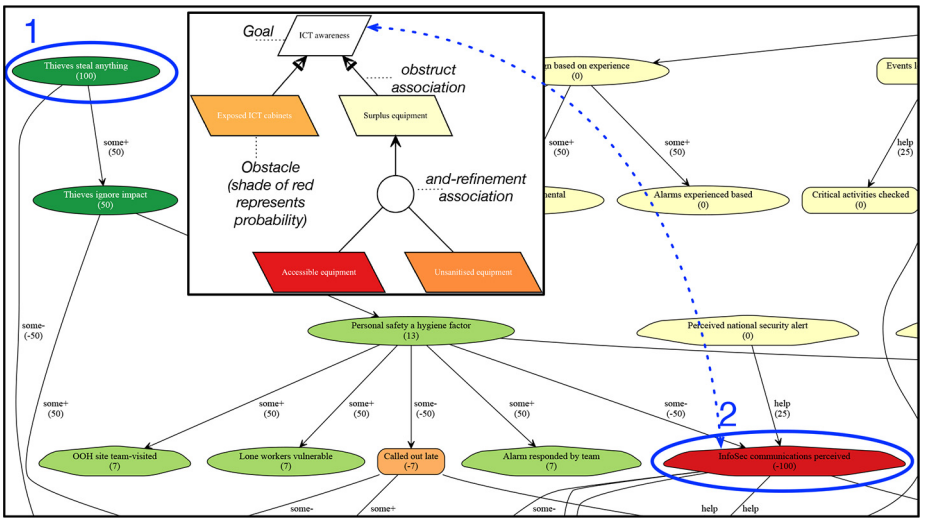


Figure 5.
Alignment between *ICT awareness* system goal in KAOS goal model (inset) and *InfoSec communications perceived* user goal in user goal model

to the interviewees for comments and, based on the feedback, subsequent axial and selective coding identified several key concepts. From this, we generated a user goal model based on Ben, we initially derived 84 user goals from both the persona characteristics and argumentation elements and 84 contribution links. The first two authors subsequently reviewed the model to remove duplicate user goals and, drawing on insights of the ethical hazards and safeguards, added additional contributions. This resulted in an additional 11 contribution links, i.e. a final model of 84 user goals and 95 contribution links.

To simulate the effect of realising each ethical hazard, we identified the user goals related to each ethical hazard and set each user goal with an appropriate level of initial satisfaction (satisfied or denied) as indicated in Table 3. Figure 7 shows the user goal model associated with Ben when the satisfaction level for these goals are set.

4.2.2 Validating ethical hazards and safeguards. To confirm the presence of the ethical hazards and safeguards, we surveyed professional penetration testers to compare whether their understanding of ethical hazards and safeguards corresponded with those in our model.

The participants recruited worked at CREST (Council of Registered Ethical Security Testers) member companies (CREST, 2021) and possessed the level of experience as those participants originally interviewed, i.e. underpinning the Ben persona. As the participants held a current CREST accreditation, they were expected to have a working knowledge of the CREST code of conduct (CREST, 2014).

The sensitivity of this topic meant that the responses received might not accurately reflect the values held by participants. For this reason, we decided to use Ben to indirectly capture participant opinions about ethical issues, in such a way that participants felt their own ethical norms and values were not being probed. To do this, we used the premortem technique to elicit responses from participants to a situation where ethical hazards were present. Premortems are a participative design technique where participants assume that a system has failed, and are asked to provide plausible reasons are given for why (Klein, 2007); previous work has shown that this technique can also be effective for providing reasons for the security exploitation of systems too (Faily *et al.*, 2012).

The pre-mortem scenario below was circulated to participants.

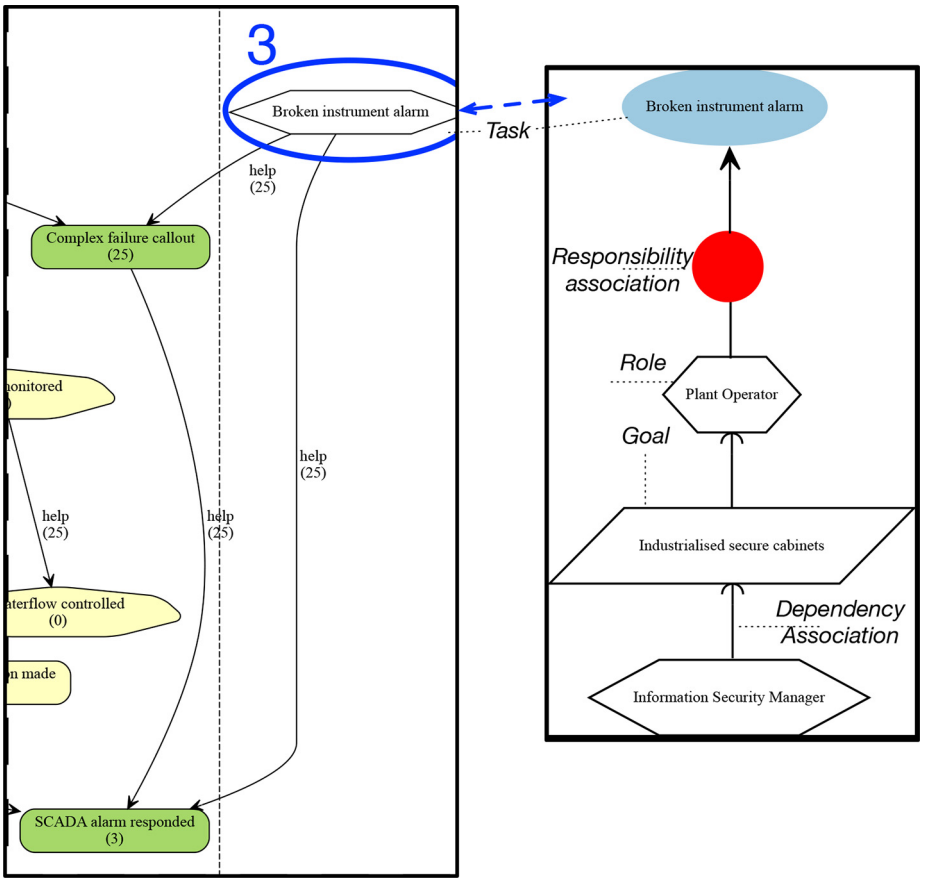


Figure 6.
Contribution of *broken instrument alarm* task to user goals (left) and related responsibility and dependency associations (right)

Hazard	Definition
Legal ambiguity	The uncertainty associated with addressing unusual forms of illegality when encountered or dilemmas between following the agreed rules of engagement or informing law enforcement agencies
Human targets	Any testing activities with the potential to jeopardise the career or well-being of test subjects
Red team vs blue team	Tensions that arise between testers (red teams) and client IT teams responsible for interacting with them (blue teams)
Client indifference	Occurrences where clients are reluctant to make changes prescribed by penetration testers or downplay the significance of problems found

Table 1.
Ethical hazards

Floods caused by Red Team

ACME Water are seeking damages from RedTeam limited liability partnership (a CREST member company) for locking a water treatment plant operator (Rick) out of their IT network. ACME considers this was the root cause of flooding that caused significant disruption in the Dorset region. As an expert witness, you have been asked to uncover evidence of unprofessional behaviour by a RedTeam engineer (Ben).

It is 1530 on a rainy Tuesday afternoon.

Ben and Alex are security engineers for RedTeam. Both are working as part of a larger project conducting a red team test on ACME Water; they are evaluating ACME's security posture to ensure their security policies on the use of IT are being followed. For this engagement, Alex is shadowing Ben. For the past week, Ben and Alex have been working out of the RedTeam office in Bournemouth but, for today only, both are working from hot desks in ACME's head office in Dorchester.

Alex is running RedTeam's ICS network auditing tool (RETINAT) to identify potential vulnerabilities on ACME's corporate IT network. For the past week, Alex has been customising RETINAT to include protocols and services used by ACME, as well as ACME phone number ranges for the tool's wardriving functionality. RETINAT was written and maintained by RedTeam staff, but the tool interfaces with various open source tools. Alex's testing with RETINAT appeared to proceed smoothly and, after writing up notes on his findings, both Ben and Alex left shortly after 5 p.m.

At 1950, Rick arrived at Moorside Water Treatment Works to start his shift. What started as light rain in the morning had now developed into a torrential downpour. Rick was worried about the potential for flooding caused by the rain, so decided to check the water level at various points across the water distribution network using the automated job system (AJS) tool. Unfortunately, after several attempts, Rick discovered that he was unable to login into Windows to access AJS. Without access to AJS, not only would Rick be unable to check the water level, but he would also be unable to control the defences available to him for reducing the water flow. Rick attempted to call the ACME IT help-desk, but was greeted with an automated reply because help is only available between 0900–1700 on weekdays.

Because he was on his own for the rest of the night, Rick attempted to call Barry – an on-call instrument technician – in the hope that he could use his credentials to access AJS. After several hours, Rick was finally able to reach Barry at 2200. Barry had been working on other jobs in an area lacking mobile phone coverage. Barry provided Rick with his own Window login details over the phone, which finally allowed him to access AJS to check and control the water level. Rick discovered that the water level was so high that several downstream villages and the main rail line to London was now flooded. The flooding subsequently disrupted transport links to the region for nearly 24 h, causing over £1m of property damage.

A review of the incident by ACME concluded that Rick was unable to login because his Windows account was locked due to an excessive number of login attempts; these login attempts appear to have been made by RETINAT. The review concluded that, had the upstream water levels been reduced by 2030, much of the downstream flooding could have been avoided. The review also noted the additional factors:

- Alex was a recent graduate. He had joined RedTeam a few months before the engagement, and started work on the ACME project the previous week.
- ACME were aware that vulnerability evaluation tests would be taking place during the week, and these tests would entail the enumeration of discovered network applications during working hours.
- Ben examined the test results before going home on Tuesday. He noted nothing unusual that would warrant client contact.
- When George – ACME's IT manager – spoke to Ben about the account lockout on Wednesday morning, Ben stated that RedTeam did not do anything that was not within their pre-agreed scope of activities.
- When the scope of activities was initially agreed, ACME indicated that several legacy applications associated with water flood level monitoring was prone to unpredictable behaviour. Ben confirmed that these applications would not be considered within the test's scope.
- Ben accepted that his relationship with George was tenuous. Several times during the past week, George spoke to Ben about problems accessing network services, which, it was claimed, was attributed to RedTeam's testing. Ben was used to receiving complaints from George, and did not think his conversation on Wednesday morning was out of the ordinary.

This scenario intentionally embodied the realisation of each ethical hazard via the appropriate satisfaction or denial of related user goals in Table 3. For example, the first factor from the review embodies denial of the *Ethical knowledge shared* user goal because Alex's recent arrival provided little opportunity for him to be onboarded with the expected ethical norms.

Participants were asked to provide open-ended responses with reasons why Ben might have behaved unethically (ethical hazards), together with things that the company could do to address the problems found (safeguards).

Five participants responded via email with 18 candidate ethical hazards and 21 candidate safeguards. Each candidate's ethical hazard was coded based on related user goals and user goals directly harmed as a result. Similarly, each candidate safeguard was categorised based on related user goals and user goals directly safeguarded. These responses and solutions are detailed in Tables 4 and 5.

The results indicated that 14 of the 18 candidate ethical hazards corresponded with at least one ethical hazard. Three of the candidate ethical hazards not in the model related to

Safeguard	Definition
Risk articulation	The explanation of security risks and the impact these have when put in a meaningful context
Service comprehension	The understanding that clients have about the penetration test service they have commissioned
Responsibility to practice	The sense of responsibility that testers have to the penetration testing profession

Table 2.
Ethical safeguards

the unwarranted trust placed in the tools used by the junior tester; the other was attributed to Ben not properly supervising the junior tester.

Whilst all candidate safeguards corresponded to at least one safeguard from the model, it is less clear how effective the proposed responses are at addressing the ethical hazards. For example, Response 2 corresponds with the satisfaction of the *Professional test run*, which implies that the *Professional standards maintained* user goal would also be satisfied. This corresponds with *Risk Articulation* and *Responsibility to Practice* safeguards but, as Figure 8 indicates, the latter goal remains denied due to the satisfaction level of user goals related to the ethical hazards. As such, whilst the safeguard is valid, it is unlikely to be effective without some combination of other safeguards being present too.

5. Discussion and limitations

The examples illustrate how our approach complements pre-existing methods for eliciting and exploring value tensions. For example, Friedman and Hendry (2019) propose the idea of *value personas* that encapsulate key values and different value tensions both within and between other personas. If such values inform the elicitation of subsequently analysed data, social goal models can then be used to explore the strengthening and weakening of different tensions related to areas such as security, privacy and trust. Moreover, the application of *value scenarios* (ibid) is analogous to our use of premortems in Example 2.

Our approach also illustrates the ease that social goal models could, potentially, be validated. The goal-oriented RE community has often use goal models to validate designs and systems, e.g. evaluate conflicting access to resources of deadline because of mutual reliance (Ali *et al.*, 2013), but work on validating the goal models themselves has relied on the arguments underpinning the model elements (Jureta *et al.*, 2008) and the mapping of goal models to natural language, which is then validated (Hassine and Amyot, 2016). Such validation approaches entail additional effort in addition to constructing the model, e.g. in specifying further relations and constructs and also in interacting with the tools to provide further input and confirmation of detected relationships and conflicts. Deriving goal models from persona cases means some confidence in the empirical basis can be gleaned both before goal models are created, and independent of the team constructing the goal model. The examples also illustrate the different ways that integrated tool-support make it possible to validate the goal model for soundness, and the impact that other models could have on goal models.

Whilst important for validating requirements, traceability is a weakness of languages such as *i** due to a lack of guidelines for working with complementary models (Pastor *et al.*, 2011). Our approach addresses this traceability problem by drawing user goal relationships from the qualitative data analysis underpinning personas. However, a limitation of our approach is the restricted expressiveness of the generated user goal models, particularly the lack of support for *strategic dependencies* between user goals. Supporting dependencies between user goals may appear trivial from a modelling perspective, but retaining

Table 3.
Ethical hazards and related user goals (and satisfaction state)

Ethical hazard	User goals (satisfied/denied)
Legal ambiguity	Sensitised to legal issues (denied), ethical knowledge shared (denied)
Human targets	Methodology explained (denied)
Red vs blue teams	Clients appraised of test details (denied), expected behaviour confirmed (denied), IT team communications respectful (denied)
Client indifference	Client expertise acknowledged (denied), indifferent (satisfied)

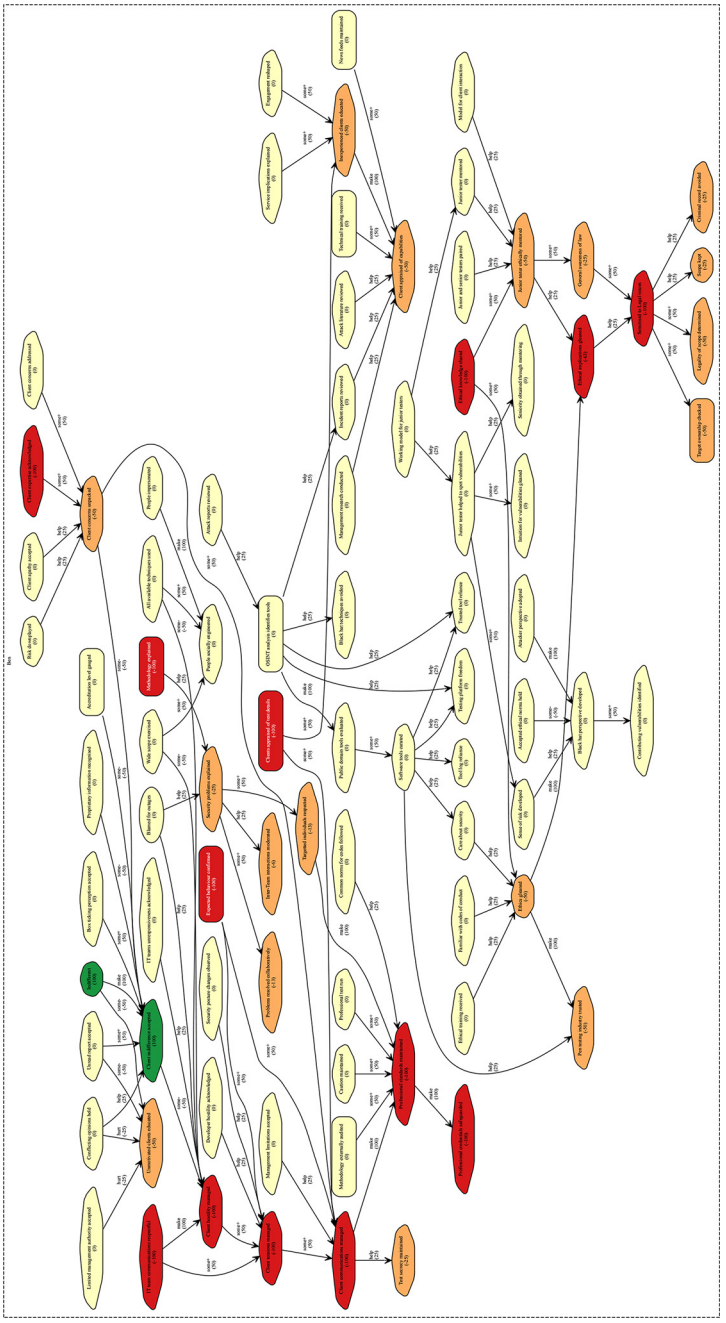


Figure 7.
CAIRIS user goal model based on ben persona

Table 4.
Pre-mortem reasons and ethical hazards

Response	Reason	Related user goals	Denied user goals	Legal ambiguity	Human targets	Red vs blue teams	Client indifference
1	He should not have used the tool if he did not fully understand the implications	Trusted tool reliance	None				
2	Ben did not review Alex's work fully. He should have identified the potential error. If he had, Ben should have also checked with the organisations IT team if they could reverse the accounts that had been locked	Junior and senior testers paired, model for client interaction, junior tester mentored, client expertise acknowledged	Ethical knowledge shared, client expertise acknowledged	X			X
3	Ben's scope had been poorly defined. There are a number of ways to test passwords and if this was something that was important to the client a different approach could have been taken, preventing accounts from being locked	Wide scope exercises, legality of scope determined, scope kept	Legality of scope determined, scope kept	X			
4	Ben's response to the clients IT department was not professional and he should have investigated the complaint	Security problems explained, IT team communications respectful, client expertise acknowledged	IT team communications respectful, security problems explained, client expertise acknowledged				X
5	Should not dismissa client's concerns	IT team communications respectful, client expertise acknowledged	IT team communications respectful, security problems explained, client expertise acknowledged				X
6	Should not bruteforce account credentials without consent and without explaining consequences	Methodology explained, client appraised of test details	Methodology explained, client appraised of test details		X		
7	Should not run a tool if you do not know what it is doing	Trusted tool reliance	None				

(*continued*)

Table 4.

Response	Reason	Related user goals	Denied user goals	Legal ambiguity	Human targets	Red vs blue teams	Client indifference
8	Legacy applications related to flood control were out of scope	Wide scope exercises, legality of scope determined, scope kept	Legality of scope determined, scope kept	X			
9	Alex should have noted accounts being locked out and should have conveyed this to Ben	Junior and senior testers paired	Junior tester ethically mentored				
10	Brute force attempts should always be performed manually. Why is his tool automating this?	Trusted tool reliance	None				
11	It is not all within Ben's control as the client should advice that the critical systems are also using domain authentication	Clients appraised of test details, client communications managed	Clients appraised of test details, client communications managed			X	
12	Ben should be more careful in a critical infrastructure scenario	Care about security	Ethics gleaned, ethical implications gleaned, sensitised to legal issues	X			
13	Ben allowed Alex, who was shadowing him, to run tools without understanding what they were doing	Junior and senior testers paired, junior tester mentored, trusted tool reliance	Ethical knowledge shared	X			
14	Ben should not have performed any testing that might lock out an account, such as password brute forcing, without first checking with the client	Clients appraised of test details, client communications managed, methodology explained	Clients appraised of test details, client communications managed, methodology explained		X	X	
15	Some legacy applications were out of scope. Even though Ben may not have tested them directly, he should have made it clear that they might be affected by testing the corporate domain	Clients appraised of test details, client communications managed, methodology explained	Clients appraised of test details, client communications managed, methodology explained		X	X	

(*continued*)

Table 4.

Response	Reason	Related user goals	Denied user goals	Legal ambiguity	Human targets	Red vs blue teams	Client indifference
16	Ben had received complaints regarding availability of services on the network. He should have followed up on these to understand whether they were caused by his actions	Clients appraised of test details, client communications managed, client expertise acknowledged	Clients appraised of test details, client communications managed, methodology explained		X	X	
17	Essentially, I believe this comes down to a lack of communication with the client before running potentially harmful tools	Clients appraised of test details, client communications managed, methodology explained	Clients appraised of test details, client communications managed, methodology explained		X	X	
18	The key information needed to determine the root-cause and attribution are whether or not RedTeam had actually performed login attempts on individual accounts (e.g. Rick's) and also whether or not that activity was part of the scope of work as had been agreed between RedTeam and ACME Water	Wide scope exercises, legality of scope determined, scope kept	Legality of scope determined, scope kept	X			

Table 5. Pre-mortem solutions and ethical safeguards

Response	Solution	Related goals	Safeguarded goals	Risk articulation	Service comprehension	Responsibility to practice
1	Review Ben's understanding of the tool used and, perhaps, arrange further training	Technical training received	Client appraised of capabilities	X	X	X
2	Add in a QA step to review scopes and assess if they have been well-defined	Professional test run	Professional standards maintained	X		X
3	Work with Ben to develop his approach when handling complaints from clients IT departments	Client communications managed, IT team communications respectful	Client communications managed, IT team communications respectful	X		
4	Highlight to Ben the responsibility that comes with mentoring	Ethical knowledge shared, junior tester ethically mentored	Ethical knowledge shared, junior tester ethically mentored			X
5	Use checklists to ensure meeting engagement criteria. Amongst these points "am I operating within scope?"	Professional test run	Professional standards maintained			X
6	The client needs to be transparent so that Ben can help them audit security	Client communications managed, IT team communications respectful	Client communications managed, IT team communications respectful	X		
7	The scope needs to be more specific to ensure that Ben knows, which accounts are in scope as opposed to just systems. Engineers should be in scope for testing also as they often have weak passwords	Scope kept, Wide scope exercised	Client hostility managed, client tensions managed			X
8	Ben should be provided with an acceptable means to identify critical accounts so as to not lock them out	Technical training received, software tools curated	Client appraised of capabilities, pen testing industry trusted		X	X
9	Red team is a vaguely used term and means something different everywhere. Ben needs to define it and ensure that the client knows what it means to perform a red team engagement and have it signed off	Methodology explained, service implications explained, client appraised of test details	Methodology explained, inexperienced clients educated, client appraised of test details	X	X	

(*continued*)

Table 5.

Response	Solution	Related goals	Safeguarded goals	Risk articulation	Service comprehension	Responsibility to practice
10	Ben should not have let Alex run tools, he should only watch. There is a reason why there is a technical client contact and Ben should work closely with him to coordinate attacks so that these systems can be monitored to ensure it is stable	Ethical knowledge shared, junior tester ethically mentored	Ethical knowledge shared, junior tester ethically mentored			X
11	Ben should have consulted with the client prior to brute forcing for account passwords across the windows domain. Although there are is time pressure during a test, staying safe ensures that his service remains professional	Methodology explained, service implications explained, client appraised of test details	Methodology explained, inexperienced clients educated, client appraised of test details	X		X
12	Why are ACME using an ICS tool to assess the network security of the corporate IT network? On another note, why are their critical water systems on the same domain as the corporate network? I think ACME should use this as an opportunity to consider secure redesign of the network	Engagement reshaped	Inexperienced clients educated			X
13	Overall, because of the nature of testing, we should defend Ben. However, if the environment was so critical, he should not have let Alex run tools, he should only watch. There is a reason why there is a technical client contact and Ben should work closely with him to coordinate attacks so that these systems can be monitored to ensure its stable	Ethical knowledge shared, junior tester ethically mentored	Ethical knowledge shared, junior tester ethically mentored			X
14	Ensure that consultants understand what their tools are doing	Technical training received	Client appraised of capabilities		X	
15	Ensure that consultants have sight of what juniors, who are shadowing, are doing	Ethical knowledge shared, junior tester ethically mentored	Ethical knowledge shared, junior tester ethically mentored			X

(*continued*)

Table 5.

Response	Solution	Related goals	Safeguarded goals	Risk articulation	Service comprehension	Responsibility to practice
16	Contact the client to gain permission to run potentially harmful tools (e.g. password brute-forcing)	Methodology explained, service implications explained, client appraised of test details	Methodology explained, inexperienced clients educated, client appraised of test details	X		X
17	During scoping, ensure that consultants understand, which "out-of-scope" systems are dependent on systems being tested, and to what degree – might they be affected?	Scope kept, wide scope exercised	Client hostility managed, client tensions managed, client communications managed			X
18	Keep logs of all testing, and follow up on complaints from clients around availability of network services. Ensure that the client has a point of contact for complaints	Client concerns addressed, tool log reliance	Client concerns unpacked, client communications managed			X
19	A more detailed investigation is needed, it was not made absolutely clear whether or not the Windows account lock-out was indeed caused by the RETINAT tool (e.g. brute force attempt?). Was there any Forensic evidence to prove it?	Client concerns addressed, tool log reliance	Client concerns unpacked, client communications managed			X
20	Identity and official position of "Rick" in the context of RedTeam or the client "ACME Water" would help	Methodology explained, service implications explained, client appraised of test details	Methodology explained, inexperienced clients educated, client appraised of test details	X		X
21	Build in a step in the scoping process to capture all legacy applications	Scope kept, wide scope exercised	Client hostility managed, client communications managed			X

traceability would necessitate changes to how the qualitative data grounding personas is elicited and analysed to ensure both personas and their collaborative aspects are encapsulated. Approaches for creating such personas already exist (Matthews *et al.*, 2011) and could provide a grounding for subsequent modelling of user goal dependencies.

Another limitation of our work is that our case study considers only individual personas. Emergent properties could exist when multiple personas collaborate and modelling and reasoning would needs to account for this emergence, e.g. through simulation and role-playing techniques at the design stages and also through feedback loops during the operation of systems. However, our initial results developing and evaluating the changes to CAIRIS indicate that user goal models place little additional performance burden to model validation checks. Because CAIRIS can incrementally import models that overlay existing models, it is possible to incrementally add personas to a baseline system to explore the impact of different personas interacting with each other. Based on the process and performance of the tool support, we believe our approach scales to multiple personas too, but a more thorough performance evaluation will be the subject of future work.

6. Conclusion

This paper presented an approach for reframing personas as social goal models and, in doing so, using both the reframed and related models to find security tensions. As a result, we have made two contributions. Firstly, we demonstrated how the user research used to construct personas can be leveraged to partially automate the construction of social goal models. Such user goals could be elicited either whilst constructing personas or afterwards – in which case the process of constructing the user goal models helps further validate the personas and the data upon which they are based. Secondly, we illustrated how minimal contributions to existing tool-support and the use of complementary design techniques facilitates automation for both the identification of implicit vulnerabilities from user goal models, and the validation of existing system goal obstructions based on user goals and user goal contributions. Our intention is not to replace traditional RE approaches to system and social goal modelling, but to show how applying them in a different way can identify and confirm potential security problems that might have otherwise remained hidden.

Future work will further examine persona characteristics and goal and task attributes to evaluate fitness between persona and actors in goal models. For example, some goals might require long-term attention span whilst others require different social skills. The user model associated with these attributes will be then used to simulate how different personas interact and whether this leads to insecurity. We will also investigate collaborative information gathering techniques to capture goal models and their personas, e.g. through an interactive

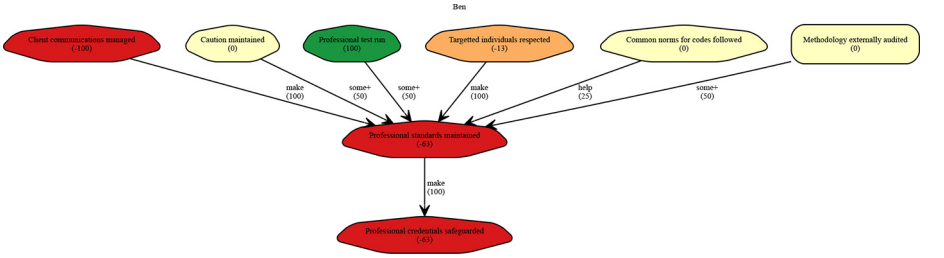


Figure 8.
User goals related to *professional standards maintained* user goal

algorithm driven by representative users providing satisfaction and denial weights and propagation options.

Note

1. CAIRIS models for both examples are available at: https://doi.org/10.5281/zenodo.4584619

References

Ali, R., Dalpiaz, F. and Giorgini, P. (2013), "Reasoning with contextual requirements: detecting inconsistency and conflicts", *Information and Software Technology*, Vol. 55 No. 1, pp. 35-57.

Amyot, D., Ghanavati, S., Horkoff, J., Mussbacher, G., Peyton, L. and Yu, E. (2010), "Evaluating goal models within the goal-oriented requirement language", *International Journal of Intelligent Systems*, Vol. 25 No. 8, pp. 841-877.

Association for Computer Machinery (2018), "ACM code of ethics and professional conduct".

AT&T (2020), "Graphviz web site", available at: www.graphviz.org

Burge, J.E., Carroll, J.M., McCall, R. and Mistrik, I. (2008), *Rationale-Based Software Engineering*, Springer.

Chapman, C.N. and Milham, R.P. (2006), "The persona's new clothes: methodological and practical arguments against a popular method", *Proceedings of the Human Factors and Ergonomics Society 50th Annual Meeting*, pp. 634-636.

Cleland-Huang, J. (2013), "Meet Elaine: a persona-driven approach to exploring architecturally significant requirements", *IEEE Software*, Vol. 30 No. 4, pp. 18-21.

Cooper, A., Reimann, R., Cronin, D. and Noessel, C. (2014), *About Face: The Essentials of Interaction Design*, John Wiley and Sons.

Corbin, J.M. and Strauss, A.L. (2008), *Basics of Qualitative Research: techniques and Procedures for Developing Grounded Theory*, 3rd ed., Sage Publications, Inc.

CREST (2014), "Code of conduct for CREST qualified individuals", available at: www.crest-approved.org/wp-content/uploads/1401-Code-of-Conduct-Individual-v4.0.pdf

CREST (2021), "CREST: Accredited companies providing penetration testing", available at: https://service-selection-platform.crest-approved.org/accredited_companies/penetration_testing/

Elahi, G., Yu, E. and Zannone, N. (2010), "A vulnerability-centric requirements engineering framework: analyzing security attacks, countermeasures, and requirements based on vulnerabilities", *Requirements Engineering*, Vol. 15 No. 1, pp. 41-62.

Faily, S. (2011), "Bridging User-Centered design and requirements engineering with GRL and persona cases", *Proceedings of the 5th International i* Workshop*, pp. 114-119, CEUR Workshop Proceedings.

Faily, S. (2018), *Designing Usable and Secure Software with IRIS and CAIRIS*, Springer.

Faily, S. and Fléchais, I. (2010), "Barry is not the weakest link: eliciting secure system requirements with personas", *Proceedings of the 24th BCS Interaction Specialist Group Conference*, BCS, pp. 124-132.

Faily, S. and Fléchais, I. (2011), "Persona cases: a technique for grounding personas", *Proceedings of the 29th ACM CHI Conference on Human Factors in Computing Systems*, ACM, pp. 2267-2270.

Faily, S. and Fléchais, I. (2014), "Eliciting and visualising trust expectations using persona trust characteristics and goal models", *Proceedings of the 6th International Workshop on Social Software Engineering*, pp. 17-24. ACM.

Faily, S., Iacob, C., Ali, R. and Ki-Aries, D. (2020a), "Identifying implicit vulnerabilities through personas as goal models", in Katsikas, S., Cuppens, F., Cuppens, N., Lambrinoudakis, C.,

Kalloniatis, C., Mylopoulos, J., Antón, A., Gritzalis, S., Meng, W. and Furnell, S. (Eds), *Computer Security*, Springer, pp. 185-202.

Faily, S., Scandariato, R., Shostack, A., Sion, L. and Ki-Aries, D. (2020b), "Contextualisation of data flow diagrams for security analysis", in Eades, H., III and Gadyatskaya, O. (Eds), *Graphical Models for Security*, Springer International Publishing, pp. 186-197.

Faily, S., Lyle, J. and Parkin, S. (2012), "Secure system? Challenge accepted: finding and resolving security failures using security premortems", *Designing Interactive Secure Systems: Workshop at British HCI*, Vol. 2012.

Faily, S., McAlaney, J. and Iacob, C. (2015), "Ethical dilemmas and dimensions in penetration testing", *Proceedings of the 9th International Symposium on Human Aspects of Information Security and Assurance*, University of Plymouth, pp. 233-242.

Friedman, B. and Hendry, D.G. (2019), *Value-Sensitive Design: Shaping Technology with Moral Imagination*, MIT Press.

Friess, E. (2012), "Personas and decision making in the design process: an ethnographic case study", *Proceedings of the 30th ACM CHI Conference on Human Factors in Computing Systems*, *ACM*, pp. 1209-1218.

Giorgini, P., Massacci, F., Mylopoulos, J. and Zannone, N. (2005), "Modeling security requirements through ownership, permission and delegation", *13th IEEE International Conference on Requirements Engineering*, pp. 167-176.

Giorgini, P., Mylopoulos, J., Nicchiarelli, E. and Sebastiani, R. (2003), "Reasoning with goal models", *Conceptual Modeling – ER 2002*, Springer, pp. 167-181.

Hassine, J. and Amyot, D. (2016), "A questionnaire-based survey methodology for systematically validating goal-oriented models", *Requirements Engineering*, Vol. 21 No. 2, pp. 285-308.

Jureta, I.J., Faulkner, S. and Schobbens, P.Y. (2008), "Clear justification of modeling decisions for goal-oriented requirements engineering", *Requirements Engineering*, Vol. 13 No. 2, pp. 87-115.

Klein, G. (2007), "Performing a project Premortem", *Harvard Business Review*, Vol. 85 No. 9, pp. 18-19.

Liu, L., Yu, E. and Mylopoulos, J. (2003), "Security and privacy requirements analysis within a social setting", *Proceedings of the 11th IEEE International Requirements Engineering Conference*, pp. 151-161.

Massacci, F. and Zannone, N. (2011), "Detecting conflicts between functional and security requirements with secure tropos: John Rusnak and the allied Irish bank", in Yu, E., Giorgini, P., Maiden, N. and Mylopoulos, J. (Eds), *Social Modeling for Requirements Engineering*, MIT Press, pp. 337-362.

Matthews, T., Whittaker, S., Moran, T.P. and Yuen, S. (2011), "Collaboration personas: a new approach to designing workplace collaboration tools", *Proceedings of the 29th ACM CHI Conference on Human Factors in Computing Systems*, pp. 2247-2256.

Mead, N., Shull, F., Spears, J., Heibl, S., Weber, S. and Cleland-Huang, J. (2017), "Crowd sourcing the creation of personae non gratae for requirements-phase threat modeling", *Proceedings of the 25th International Requirements Engineering Conference*, pp. 412-417.

Moody, D.L., Heymans, P. and Matulevicius, R. (2009), "Improving the effectiveness of visual representations in requirements engineering: an evaluation of i* visual syntax", *Proceedings of the 17th IEEE International Requirements Engineering Conference*, IEEE, pp. 171-180.

Mouratidis, H. and Giorgini, P. (2007), "Secure tropos: a security-oriented extension of the tropos methodology", *International Journal of Software Engineering and Knowledge Engineering*, Vol. 17 No. 2, pp. 285-309.

Mouton, F., Malan, M.M., Kimppa, K.K. and Venter, H. (2015), "Necessity for ethics in social engineering research", *Computers and Security*, Vol. 55, pp. 114-127.

Nunes Rodrigues, G., Joel Tavares, C., Watanabe, N., Alves, C. and Ali, R. (2018), "A persona-based modelling for contextual requirements", in Kamsties, E., Horkoff, J. and Dalpiaz, F. (Eds), *Requirements Engineering: Foundation for Software Quality*, Springer, pp. 352-368.

Paja, E., Dalpiaz, F. and Giorgini, P. (2013), "Designing secure socio-technical systems with STS-ml", *Proceedings of the 6th International i* Workshop 2013*, pp. 79-84.

Pastor, O., Estrada, H. and Martínez, A. (2011), "Strengths and weaknesses of the i* framework: an empirical evaluation", in Yu, E., Giorgini, P., Maiden, N. and Mylopoulos, J. (Eds), *Social Modeling for Requirements Engineering*, MIT Press, pp. 607-643.

Pendse, S.G. (2011), "Ethical hazards: a motive, means, and opportunity approach for curbing corporate unethical behavior", *Journal of Business Ethics*, Vol. 107 No. 3, pp. 265-279.

Regev, G. and Wegmann, A. (2005), "Where do goals come from: the underlying principles of goal-oriented requirements engineering", *13th IEEE International Conference on Requirements Engineering*, pp. 353-362.

Simon, H.A. (1979), "Rational decision making in business organizations", *The American Economic Review*, Vol. 69 No. 4, pp. 493-513.

Sindre, G. and Opdahl, A.L. (2007), *Capturing Dependability Threats in Conceptual Modelling*, Springer.

Toulmin, S. (2003), *The Uses of Argument*, updated ed., Cambridge University Press.

van Lamsweerde, A. (2009), *Requirements Engineering: From System Goals to UML Models to Software Specifications*, John Wiley and Sons.

Yu, E. (1997), "Towards modeling and reasoning support for early-phase requirements engineering", *Proceedings of the 3rd IEEE International Symposium on Requirements Engineering*, IEEE, pp. 226-235.

Yu, E.S. (2009), "Social modeling and i*", *Conceptual Modeling: Foundations and Applications: Essays in Honor of John Mylopoulos*, Springer, pp. 99-121.

Yu, E., Giorgini, P., Maiden, N. and Mylopoulos, J. (2011), "Social modeling for requirements engineering: an introduction", in Yu, E. (Ed.), *Social Modeling for Requirements Engineering*, MIT Press.

Yu, E. (1995), "Modeling strategic relationships for process reengineering", PhD thesis, University of Toronto.

About the authors

Dr Shamal Faily is a Principal Lecturer in Systems Security Engineering at Bournemouth University. He completed his DPhil in Computer Science at the University of Oxford in 2011. Prior to his doctoral research, he was a software engineer within Logica's Space business. Shamal Faily is the corresponding author and can be contacted at: sfaily@bournemouth.ac.uk

Dr Claudia Iacob is a Senior Lecturer in Computer Science at the University of Portsmouth, UK. She earned her PhD in Computer Science from the University of Milan. Her research focussed on user generated online content and its role in software development, as well as software engineering education and training.

Professor Raian Ali is a Professor at the Information and Computing Technology Division, College of Science and Engineering, Hamad Bin Khalifa University, Qatar. His research focusses on the inter-relation between technology design and social requirements such as motivation, transparency, well-being and responsibility.

Dr Duncan Ki-Aries is a Lecturer in Computer Science and Security at Bournemouth University. Previous research completed in his PhD was specifically focussed towards Security Risk Assessment in Systems of Systems, whilst integrating the use of tool-support with CAIRIS.