Analysing and Visualising Security and Usability in IRIS

Shamal Faily
Computing Laboratory
University of Oxford
Oxford OX1 6UD, UK
shamal.faily@comlab.ox.ac.uk

Ivan Fléchais
Computing Laboratory
University of Oxford
Oxford OX1 3QD, UK
ivan.flechais@comlab.ox.ac.uk

***Abstract*—Despite a long standing need to incorporate human factors into security risk analysis, taking a balanced approach to analysing security and usability concerns remains a challenge. Balancing security and usability is difficult due to human biases in security perception, and managing the sheer volume of data arising from risk and task analysis. This paper presents an approach for qualitatively and quantitively analysing and visualising the results of risk and task analysis. We demonstrate this approach using a realistic example, and we discuss how these techniques fit within the larger context of secure systems design.**

I. Introduction

Risk analysis is a powerful tool for reasoning about the trade-offs involved in secure systems design, but there is a long standing need to incorporate human factors into this process. Unfortunately, risk and usability ratings for a system design are sufficiently coloured by analyst perceptions that human error can easily creep into any valuation. Without a means to rapidly score and visualise the current state of analysis, the security-usability balance can become uneven as risk analysis becomes more advanced.

To many engineers, usability is synonymous with user interface design [1]. Usability is, however, more than just about designing interfaces, it is a quality concerning the people interacting with these interfaces and how they use them to perform tasks. Recent work suggests that enforcing usability should be a common responsibility during analysis activities [2], but there are currently no tools, conceptual or otherwise, which measure the impact to usability of secure design decisions; without this information, enforcing responsibility is difficult as the sole focus will be on mitigating risks.

In [3], we introduced IRIS (Integrating Requirements and Information Security), an integrated framework for usable and secure software engineering. This paper discusses the specifics of how risk and usability are analysed within IRIS, and how the results of this analysis can be visualised. Section II describes related work on analysing and visualising risk and usability, before sections III and IV describe the process used by IRIS for qualitatively rating and quantitatively scoring risks and tasks. Section V describes how these ratings and scores are visualised in IRIS, before presenting an example of this approach in section VI. Finally, in section VII, we discuss how the techniques described in this paper can be applied within the larger context of secure systems design.

II. Related work

To many people, 'risk' is synonymous with some form of threat, danger or hazard. Techno-scientific approaches to risk expand on this synonym by incorporating the possibility of harm occurring; for example, Bradbury [4] defines risk as "the product of the probability and consequences (the magnitude and severity) of an adverse event". By extension, risk analysis is contingent on data about the likelihood of danger, and the impact this might have on a context of study. However, a weakness of existing quantitative approaches to risk analysis is that the nature of risk may become lost in precisely this data. When we think of an attack, we may consider the system as a whole, rather that subsystems of particular interest to an attacker or a defender. When stakeholders contribute data as part of a risk analysis exercise, they are likely to value particular assets over others. Moreover, when considering the protection of assets, or threats to them, stakeholders think in terms of particular properties which need to be safeguarded. Therefore, a framework for quantitative risk analysis data needs to be sensitive to the values participants place on assets, as well as security properties associated with threats, vulnerabilities and risk mitigation approaches.

Our work involves understanding how risk analysis can help or hinder the usability of work carried out by different people. Task analysis is concerned with the study of work performance, and aims to model how work, hereafter known as 'tasks', can be used to change the application domain [5]. Although different representations for task analysis exist, the most prevalent mechanism used for describing tasks in both HCI and Secure Software Engineering is the scenario, a textual narrative which describes how a stakeholder interacts with the domain under study. It has, however, been argued that scenarios make little reference to the stakeholders being described within them [6]. As introducing a security control impacts usability for people performing the task, rather than just the task per se, humanising scenarios is important. One

means of capturing this human dimension is augmenting scenarios with personas; these provide a descriptive model of how archetypical users behave, think, what they wish to accomplish, and why [7]. By combining personas and scenarios, not only do we avoid introducing undue assumptions into analysis, we can realistically categorise the usability of tasks with respect to their indicative users.

Before stakeholders can measure the impact of usability of secure system design decisions, they need to identify the impact itself. Although there appears to be little work on visualising task and usability metrics, techniques from information visualisation have been applied to security risk analysis [8] [9], and risk analysis as part of the design process [10]. Hogganvik's recent thesis on the subject of visualising risk analysis [11] looked at how the colour and shape of model elements can be used to make information more accessible. Results from this work concluded that parsimony is important with respect to the number of model symbols and colours employed, and that colour may be a useful means of distinguishing the value of different risks.

III. Risk Analysis

Score	Threat Likelihood
0	Incredible
1	Improbable
2	Remote
3	Occasional
4	Probable
5	Frequent

Score	Vulnerability Severity
0	Negligible
1	Marginal
2	Critical
3	Catastrophic
4	Frequent

Score	Risk Rating
1	Intolerable
2	Undesirable
3	Tolerable
4	Negligible

	Consequence			
Frequency	**Catastrophic**	**Critical**	**Marginal**	**Negligible**
Frequent	1	1	1	2
Probable	1	1	2	3
Occasional	1	2	3	3
Remote	2	3	3	4
Improbable	3	3	4	4
Incredible	4	4	4	4

Figure 1. IEC 61508 Tables for Threat Likelihood, Vulnerability Severity, and Risk Categorisation

A risk rating can be assigned based on likelihood and severity tables in IEC 61508 [12] (see figure 1). However, this rating does not reflect values held about individual assets or threats. To score risks with respect to the perceived value of the assets threatened, we define a security property as a row vector $\begin{bmatrix} c & i & a & o \end{bmatrix}$, where c, i, a, and o represent the values held for confidentiality, integrity, availability and accountability respectively. Each element n is valued $0 \leq n \leq 3$ based on whether the value held for that element is none, low, medium or high. The likelihood of the threat being realised, L_r is computed using the equation

$$L_r = L_t - \bar{m_t}$$

where L_t = the likelihood of the threat t associated with risk r, and $\bar{m_t}$ is the mean likelihood value for the set of countermeasures mitigating the likelihood of L_t occurring. The values of L_t and $\bar{m_t}$ exist within the range $0 \leq n \leq 5$, and map to the likelihood categories in figure 1. The severity of the vulnerability exposed by risk r is computed using the equation

$$S_r = S_v - \bar{m_s}$$

where S_v = the severity of the vulnerability v associated with risk r, and $\bar{m_s}$ is the mean severity for the set of countermeasures mitigating the severity of S_v. Like threat severity, vulnerability values exist within the range $0 \leq n \leq 4$ and map to the vulnerability categories in figure 1.

Risk impact is described by a security property, representing the values held in the assets at risk from risk r. Risk impact is computed using the equation

$$P_r = (P_t \times P_a) - \bar{m_p}$$

where P_t = the security properties of the threat associated with risk r, P_a = the security properties of the vulnerable or threatened assets at risk, and $\bar{m_p}$ = the mean security properties for the countermeasures targeting the risk's threat or vulnerability.

Finally, the calculation for the Risk Score of risk r, R_r, is computed, as the product of the threat likelihood, the severity of the vulnerability, and the risk impact to the threatened assets.

$$R_r = L_r \times S_r \times P_r$$

Each element of row vector is added together, and the sum is normalised to an integer between 1 and 9. If, during the above computations, negative numbers are calculated, these values are resolved to 0.

IV. Task Analysis

When defining tasks, four properties are set for each persona participating in a scenario. Each of these properties, described in figure 2 map to one of the usability components of ISO 9241-11 [13].

Each qualitative value x associated with a property maps to a natural number in the range $0 \leq x \leq 3$, which correspond to the qualitative values of None, Low, Medium, and High. To ensure equal weighting for all 3 usability components, the usability of a task U_t is computed using the equation

$$U_t = \frac{a \bar{+} b}{2} + \bar{c} + \bar{d}$$

Property	ISO 9241-11 Usability Component	Description	Values
Duration	Efficiency	The time taken by a persona to complete the task.	• None • Seconds • Minutes • Hourly or longer
Frequency	Efficiency	The frequency a persona carried out the task.	• None • Hourly or more • Daily - Weekly • Monthly or less
Demands	Satisfaction	The mental or physical demands on a persona.	• None • Low • Medium • High
Goal Conflict	Effectiveness	The degree to which the task interferes with the persona's work or personal goals.	• None • Low • Medium • High

Figure 2. Task Usability Properties

where $\frac{\bar{a+b}}{2}$ is the mean task efficiency, $\bar{c}$ is the mean task satisfaction, and $\bar{d}$ is the mean task effectiveness. Variables a, b, c, and d refer to the task duration, frequency, demands, and goal conflict respectively. The mean value is taken across all personas carrying out the task in question. In general, the higher the value of U_t, the less usable a task is for the personas associated with it. Because values like low, medium, and high are ambiguous with respect to duration and frequency, more meaningful values are used. For duration, the qualitative ratings used are *Seconds*, *Minutes*, and *Hours or Longer* are associated with the values 1,2, and 3 respectively. For frequency, the ratings used are *Monthly or less*, *Daily - Weekly*, and *Hourly or more*.

When mitigating risks, one or more roles are associated with each mitigating countermeasure; these roles will, in some way, be directly affected by the countermeasure being designed. By associating roles with countermeasure within IRIS, candidate personas and their tasks can be identified. For each task-persona pairing, countermeasure usability properties can be specified. These properties, described in figure 3 are similar, but not identical to, those specified in figure 2.

Based on this countermeasure usability data, it is possible to calculate the countermeasure usability factor C_t. The right hand side of the equation which computes C_t is identical to U_t, i.e.

$$TU_t = \frac{\bar{a+b}}{2} + \bar{c} + \bar{d}$$

however the values of are different. $\frac{\bar{a+b}}{2}$ is the mean contribution to task efficiency, $\bar{c}$ is the mean contribution to task satisfaction, and $\bar{d}$ is the mean contribution to task effectiveness. Like U_t, the variables a, b, c, and d refer to the task duration, frequency, demands, and goal conflict respectively. The mean contributing value is taken across all countermeasures affecting the task in question. Unlike U_t,

Property	ISO 9241-11 Usability Component	Description	Values
Duration	Efficiency	The degree to which the countermeasure helps or hinders the time taken by a persona to complete the task.	• High Help • Medium Help • Low Help • None • Low Hindrance • Medium Hindrance • High Hindrance
Frequency	Efficiency	The degree to which the countermeasure increases or decreases the frequency a persona needs to carry out the task.	• High Help • Medium Help • Low Help • None • Low Hindrance • Medium Hindrance • High Hindrance
Demands	Satisfaction	The degree to which the countermeasure increases or decreases the mental or physical demands on a persona while carrying out the task.	• High Help • Medium Help • Low Help • None • Low Hindrance • Medium Hindrance • High Hindrance
Goal Conflict	Effectiveness	The degree to which the task helps or hinders the persona's work or personal goals.	• High Help • Medium Help • Low Help • None • Low Hindrance • Medium Hindrance • High Hindrance

Figure 3. Countermeasure Task Usability Properties

however, each qualitative value x associated with a property maps to an integer in the range $-3 \leq x \leq 3$.

Based on these equations, we compute the task summative usability SU_t to be

$$SU_t = U_t + TU_t$$

Like U_t, the higher the score, the less usable the task is for the associated personas. After calculating U_t and SU_t, the score is normalised to a natural number in the range $0 \leq n \leq 9$. Given the potential of a task to increase or decrease usability, this value remains unchanged irrespective of it being a high positive or negative number.

V. Visualising Risk and Task Analysis

Figure 4. Risk (top, red) and Task Usability (bottom, blue) colour charts

The on-going results of risk and task analysis are visualised within IRIS using a *risk analysis* model. This model provides a quick-look view of the current analysis.

With so much quantitative and qualitative information associated with risk analysis, visual clutter can be problem as the model expands. Minimal distinctions in colour can be used to reduce visual clutter, and small contrasts enrich the visual signal increasing the number of possible distinctions

[14]. Therefore, we map the normalised values for R_r and SU_t to the respective risk and task usability colour charts in figure 4. The higher the risk or task usability score, the deeper the hue of red or blue. Threat likelihood and vulnerability severity scores map to a similar colour chart to that of risk. An example of how these colours are applied to elements on the IRIS risk analysis model is provided in figure 5.

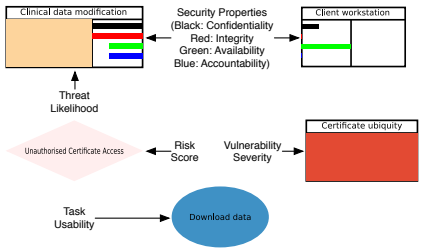

Figure 5. Sample risk analysis model nodes

As figure 5 illustrates, information about the security properties is also coded within asset and threat elements. Histograms indicate whether or not values are held for each property and, if so, whether that property is low, medium, or high. The colours selected for the confidentiality, integrity, availability, and accountability histograms are the 3 primary colours, together with black; the use of black and primary colours provide the maximum differentiation between property types [15].

VI. Example: Scoring and Visualising the Upload of Clinical Data

In this example, we describe an example of how usability and risk can be scored based on the formulae in the previous sections. Our example looks at the usability and security associated with uploading sensitive data to a computational grid. We have previously defined a task called 'upload data', which is carried out by a pre-defined persona: Alice. This task begins with Alice anonymising a clinical data set, which involves removing as much personalised data as possible while still enabling her analysis software to work. Alice then uploads this data, tagging this as available only to members of her research group. In the example, Alice spends several minutes on this task, which she usually carries out each working day. Given the persona profile, this is a low demand task which generally supports her goals.

$$\begin{aligned} U_t &= \frac{a \bar{+} b}{2} + \bar{c} + \bar{d} \\ &= \frac{2+2}{2} + 1 + 2 \\ &= 5 \end{aligned}$$

We now consider the risk of 'Unauthorised Certificate Access'. In this example, this risk is realised by a social engineer obtaining access to a client workstation and installing a user digital certificate. Such an attacker could be a journalist more interested in the newsworthiness of the attack than the value of the compromised data.

The attacker exploits a vulnerability arising from the difficulty of obtaining digital certificates. Rather than undertaking the onerous task of legitimately obtaining a digital certificate, which involves obtaining permission from line management, filling out forms, and sending several confirmatory emails to a Certificate Authority, many users instead choose share digital certificates among themselves. Such insecure behaviour arises not from malevolence, but from a desire to carry out their work without undue hindrance. The attacker exploits this 'certificate ubiquity' vulnerability.

Realising the risk might involve a journalist obtaining site access, including access to a workstation, and masquerading as a new member of staff who, frustratingly, has not been given access to all the tools she needs to carry out her work. In [16], we elaborate this attack using a Misuse Case [17] to describe how an attacker pretends to be a new post-doctoral researcher whose new supervisor happens to be away from the office when she arrives for her first day at work.

This risk can be assessed both qualitatively and quantitatively. We can evaluate the risk qualitatively based on threat likelihood and vulnerability severity. These values were occasional and critical respectively, giving rise to a rating of Undesirable.

We assess this risk quantitatively by calculating its risk score. The likelihood and severity scores mapped to 2 and 3 respectively. The threat targeted only the confidentiality of these assets, but two assets were targeted by this threat; these were the client workstation ($\begin{bmatrix}1 & 0 & 3 & 0\end{bmatrix}$) and a user certificate ($\begin{bmatrix}2 & 0 & 3 & 1\end{bmatrix}$). The asset exposed by the vulnerability is the user certificate Using this information, we calculated the risk score R_r:

$$\begin{aligned} L_r = L_t - \bar{m}_t &= 3 - 0 \\ &= 3 \\ S_r = S_v - \bar{m}_s &= 2 - 0 \\ &= 0 \\ P_r &= (P_t \times P_a) - \bar{m}_p \\ &= (\begin{bmatrix}3 & 0 & 0 & 0\end{bmatrix} \times \begin{bmatrix}3 & 0 & 6 & 1\end{bmatrix}) \\ &= \begin{bmatrix}9 & 0 & 0 & 0\end{bmatrix} \\ R_r &= L_r \times S_r \times V_r \\ &= 3 \times 2 \times \begin{bmatrix}9 & 0 & 0 & 0\end{bmatrix} \\ &= \begin{bmatrix}54 & 0 & 0 & 0\end{bmatrix} \end{aligned}$$

After rounding R_r down, the normalised score resolved to 9.

In this example, we mitigate this risk by making user certificates host based, such that an issued certificate can only be used for a given client workstation. This countermeasure is considered highly effective at targeting the certificate

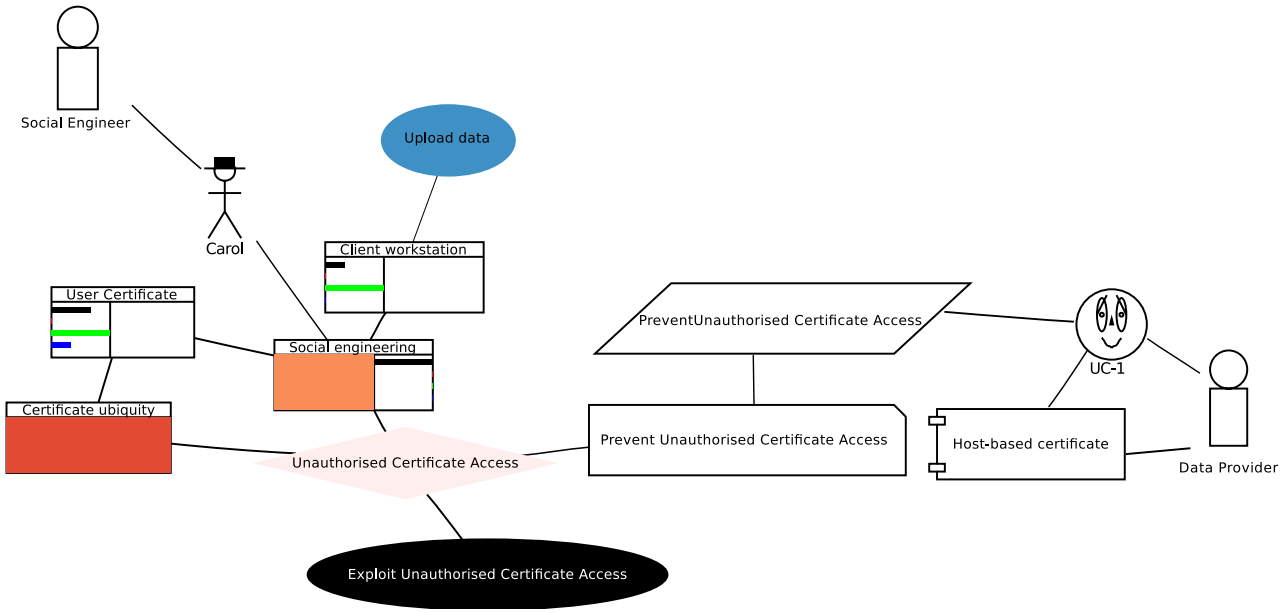

Figure 6. IRIS Risk Analysis model example

ubiquity vulnerability, motivating the value of 3 (High) for $\bar{m}_s$. This countermeasure also fosters, albeit only to a minor extent, values of confidentiality and accountability, giving rise to a score of $\begin{bmatrix}1 & 0 & 0 & 1\end{bmatrix}$ for $\bar{m}_p$. Based on this information, the risk score can be re-evaluated.

$$\begin{aligned} L_r = L_t - \bar{m}_t &= 3 - 0 \\ &= 3 \\ S_r = S_v - \bar{m}_s &= 2 - 3 \\ &= -1 \\ P_r &= (P_t \times P_a) - \bar{m}_p \\ &= (\begin{bmatrix}3 & 0 & 0 & 0\end{bmatrix} \times \begin{bmatrix}3 & 0 & 6 & 1\end{bmatrix}) - \\ &\quad \begin{bmatrix}1 & 0 & 0 & 1\end{bmatrix} \\ &= \begin{bmatrix}8 & 0 & 0 & -1\end{bmatrix} \\ R_r &= L_r \times S_r \times V_r \\ &= 3 \times 0 \times \begin{bmatrix}8 & 0 & 0 & -1\end{bmatrix} \\ &= \begin{bmatrix}0 & 0 & 0 & 0\end{bmatrix} \end{aligned}$$

These results show that while certain security properties remain threatened, the severity of the vulnerability was rendered inert, thereby reducing the risk score to the lowest possible value. As this countermeasure appeared to be effective, we chose to generate a new asset for this policy. The security properties of this new asset were based on the values placed on the countermeasure, which varied based on the contexts the new asset was situated in.

This countermeasure also impacts the usability of uploading data. Because Alice now needs to make sure she uploads data only from a particular machine, introducing this countermeasure is a slight hindrance to her other goals. As such, the summative task usability can now be evaluated.

$$\begin{aligned} SU_t &= U_t + TU_t \\ &= 5 + \frac{0+0}{2} + 0 + 1 \\ &= 6 \end{aligned}$$

As a consequence, the risk is mitigated however the task usability is slightly worsened due to the extra effort involved in complying with the new policy. A risk analysis model showing the results of this, and related analysis, in IRIS is provided in figure 6.

VII. Discussion

The techniques we have described can support the specification of security requirements, and secure systems design in general. However, these need to be implemented as part of tool-support, and this tool-support needs to be integrated into the secure systems design process. Moreover, the design process needs to capture well-grounded data to ensure analysis is reflective of the environments within which the system will operate.

IRIS framework builds upon a meta-model for integrated usability, requirements, and risk analysis, which is founded in best practice in HCI, Security Requirements Engineering, and Information Security. In related work [3], we have demonstrated tool support which implements the techniques described by this paper. Initial validation of this tool involved a retrospective analysis of a recently completed UK e-Science project.

We have devised a design process to elicit and specify the requirements of secure and usable software systems.

This process incorporates techniques from Contextual Design [18], Goal Oriented Requirements Engineering [19], Security Requirements Engineering [17], and HCI Security [20]. Applying this process involves carrying out a scoping workshop with stakeholders, followed by Contextual Interviews [18] with representative users in their work contexts. Data from these sessions are used to elicit candidate tasks, assets, threats, and vulnerabilities, and generate personas [21]. When Contextual Inquiry is complete, AEGIS risk analysis workshops [20] are held with representative stakeholders. The techniques described by the paper are used to inform security requirements and design decisions in real-time. A critical infrastructure case study evaluating this design process is on-going, and future work will share the results of this study.

VIII. Conclusion

Reasoning about security and usability is a challenge during risk analysis, not least because analyst bias and data explosion can occur when analysis becomes developed. This challenge motivates the need to rapidly analyse and visualise the impact that task and risk analysis can have on each other. In this paper has introduced techniques for analysing risks and tasks, and visualising the results of this analysis. We have illustrated this approach with a working example, and discussed how this work fits into the larger context of secure systems design.

IX. Acknowledgements

The research described in this paper was funded by EPSRC CASE Studentship R07437/CN001. We are very grateful to Qinetiq Ltd for their sponsorship of this work.

References

[1] A. Seffah and E. Metzker, "The obstacles and myths of usability and software engineering," *Commun. ACM*, vol. 47, no. 12, pp. 71–76, 2004.

[2] J. Heiskari, M. Kauppinen, M. Runonen, and T. Männistö, "Bridging the Gap Between Usability and Requirements Engineering," in *17th IEEE International Requirements Engineering Conference*. IEEE, 2009, pp. 303–308.

[3] S. Faily and I. Fléchais, "Context-Sensitive Requirements and Risk Management with IRIS," in *17th IEEE International Requirements Engineering Conference*. IEEE, 2009, pp. 379–380.

[4] J. A. Bradbury, "The policy implications of differing concepts of risk," *Science, Technology, and Human Values*, vol. 14, no. 4, pp. 380–399, 1989.

[5] D. Diaper, "Understanding Task Analysis for Human-Computer Interaction," in *The Handbook of Task Analysis for Human-Computer Interaction*, D. Diaper and N. A. Stanton, Eds. Lawrence Erlbaum Associates, 2004, pp. 5–47.

[6] N. Mikkelson and W. O. Lee, "Incorporating User Archetypes into Scenario-based Design," in *9th Annual Usability Professionals' Association (UPA) Conference, Asheville, North Carolina*, 2000.

[7] A. Cooper, R. Reimann, and D. Cronin, *About Face 3: The Essentials of Interaction Design*. Wiley, 2007.

[8] I. Hogganvik and K. Stølen, *A graphical approach to risk identification, motivated by empirical investigations*, 2006, vol. 4199 LNCS.

[9] R. Gandhi and S.-W. Lee, "Visual analytics for requirements-driven risk assessment," *Requirements Engineering Visualization, 2007. REV 2007. Second International Workshop on*, pp. 6–6, Oct. 2007.

[10] M. Feather, S. Cornford, J. Kiper, and T. Menzies, "Experiences using visualization techniques to present requirements, risks to them, and options for risk mitigation," *Requirements Engineering Visualization, 2006. REV '06. First International Workshop on*, pp. 10–10, Sept. 2006.

[11] I. Hogganvik, "A graphical approach to security risk analysis," Ph.D. dissertation, University of Oslo, 2007.

[12] *IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems. Parts 1-7*. Switzerland: International Electrotechnical Commission, 1998-2005.

[13] ISO, "ISO 9241-11. Ergonomic requirements for office work with visual display terminals (VDT)s - Part 11 Guidance on usability," Tech. Rep., 1998.

[14] E. R. Tufte, *Visual Explanations: Images and Quantities, Evidence and Narrative*. Cheshire, Connecticut: Graphics Press, 1997.

[15] ——, *Envisioning information*. Cheshire, Conn. (P.O. Box 430, Cheshire 06410): Graphics Press, 1990.

[16] S. Faily and I. Fléchais, "Usable Secure Software Engineering with IRIS," in *International Symposium on Engineering Secure Software and Systems*, ser. Submitted. Springer-Verlag, February 2010.

[17] G. Sindre and L. Opdahl, "Eliciting security requirements with misuse cases," *Requirements Engineering*, vol. 10, no. 1, pp. 34–44, 2005.

[18] H. Beyer and K. Holtzblatt, *Contextual design: defining customer-centered systems*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998.

[19] A. Van Lamsweerde and E. Letier, "Integrating obstacles in goal-driven requirements engineering," Apr 1998, pp. 53–62.

[20] I. Flechais, C. Mascolo, and M. A. Sasse, "Integrating security and usability into the requirements and design process," *International Journal of Electronic Security and Digital Forensics*, vol. 1, no. 1, pp. 12–26, 2007.

[21] 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.