

# Designing Usable and Secure Software with IRIS and CAIRIS

Shamal Faily

# Designing Usable and Secure Software with IRIS and CAIRIS

 Springer

Shamal Faily  
Department of Computing & Informatics  
Bournemouth University  
Poole, Dorset  
UK

ISBN 978-3-319-75492-5                      ISBN 978-3-319-75493-2 (eBook)  
<https://doi.org/10.1007/978-3-319-75493-2>

Library of Congress Control Number: 2018938649

© Springer International Publishing AG, part of Springer Nature 2018

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper

This Springer imprint is published by the registered company Springer International Publishing AG part of Springer Nature  
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

*To Claudia*

# Foreword

It is seldom the case in software engineering that we encounter a book that is relevant to practitioners, researchers and educators. Most books will either focus on the practitioner audience or the researcher/educator audience. This book is a welcome addition to the small set of books in our field that are relevant to all of these audiences. Regardless of which lifecycle model you use, if you are at all concerned with security and usability (and you should be!), this book is relevant to you.

I especially liked the mapping of the material in the book to its use by practitioners, researchers and educators. Since I have had all of those roles over the course of my career, I know how important it is to be able to focus on the topics of interest to me, and to be able to skim the others. Naturally, those of us who have written books would like our readers to read the entire book, but when we put ourselves in the shoes of our readers, we realise that we seldom have the time to do this.

The requirements engineering research community is vibrant but relatively small—a few hundred researchers, almost all of whom know one another. When we focus specifically on security and usability, as it relates to specification and design, the field becomes even more specific. This book is a welcome addition to the relatively small number of books that focus on security and usability in design.

The vast majority of the literature in the area consists of journal and conference research papers. The educational material tends to consist of tutorials, a few lectures, and maybe a course or two here and there. One of the significant contributions of this book is to provide an excellent source of information about security and usability in requirements and design, with copious references. It also provides a combination of overview material and more detailed explanations, depending on the subject at hand.

The first few chapters provide a nice overview of the security and usability concepts needed to understand the remainder of the book. In Chap. 3, the IRIS meta-model is introduced. I particularly liked the table with the core concepts of IRIS and the description of each. So often terminology is introduced without any explanation whatsoever, but that is not the case here. In Chap. 4, the IRIS process framework is introduced. The associated perspectives on usability, requirements

and security are all discussed, and then in Chap. 5 the tool support provided by CAIRIS is introduced. The discussion of personas and scenarios in Chap. 6 sets the stage for the case studies that follow.

The case studies help to make the methodologies more useful to practitioners, and from a practitioner perspective, tool support is essential. Paper exercises are OK for the classroom, but when large complex systems are under development, tool support is needed. Remaining chapters delve into architectural risk analysis, design issues and future research challenges.

All in all this book is an excellent choice for those who need to be concerned with security and usability, and that is just about everyone! This book is especially useful during the early lifecycle activities of requirements engineering, architecture and high-level design. In fact, given the current focus on security and usability, I think this book is a good choice for all systems and software engineers. It will expand your horizons.

Pittsburgh, PA, USA  
March 2018

Nancy R. Mead, Ph.D.  
SEI Fellow, Carnegie Mellon University  
IEEE Fellow  
ACM Distinguished Educator

# Preface

## Why do We Need This Book?

Everyone agrees that security needs to be considered as early as possible when designing any form of technology. Unfortunately, the security industry has been so effective at promoting stories about potential ‘digital Pearl Harbours’ and astronomical costs of cybercrime to national economies that many people are being paralysed into inaction, or pushed into irrationality when thinking about security. Given the demands to deliver more, sooner, with fewer resources, it can be appealing to suspend disbelief and trust that a piece of technology which promises the moon and costs the earth really will mitigate the latest threats, fit seamlessly into our enterprise's infrastructure, or otherwise ‘add value’. Alternatively, one might treat scare stories as noise, and—as Cormac Herley once wrote<sup>1</sup>—conclude that the rational thing to do is ignore any advice that has been given. Unfortunately, relying on such shortcuts means that a product or service may fail to protect things of most value to us when we need them most.

When technology fails or is at risk of failing, it can also be tempting to blame the person interacting with it, i.e. blaming the user for committing a human error. However, if security is to be built-in to technology then it needs to respect the goals and demands of those that rely on it. This means rather than treating users as the ‘weakest link’ in security, a system design should consider the characteristics of its direct or indirect users. Unfortunately, as hard as building security it can be, it appears building both usability *and* security are even harder. Platitudes like ‘you should think of security and usability up-front’ are prescribed frequently, but useful best practice or case study examples seem to be in short supply. Identifying the

---

<sup>1</sup>Herley C. So Long, and No Thanks for the Externalities: The Rational Rejection of Security Advice by Users. In: NSPW ‘09: Proceedings of the 2009 New Security Paradigms Workshop. ACM; 2009. pp. 133–144.

form that design techniques and tools might take to demonstrably incorporate security and usability into the earliest stages of software design was the main motivation for devising IRIS and CAIRIS.

## What are IRIS and CAIRIS?

IRIS (Integrating Requirements and Information Security) is a process framework that can be used to devise processes for designing usable and secure software. CAIRIS (Computer-Aided Integration of Requirements and Information Security) is a software platform for eliciting, specifying and validating secure and usable systems. It was built from the ground up to support IRIS, and includes the elements necessary for usability, requirements and risk analysis.

The book contains everything you need to understand what IRIS and CAIRIS are, and how you can use the two to design usable and secure software at an early stage. However, despite the book's title, this book is more than just a playbook for IRIS, or a manual for CAIRIS. The book was written to show how design activities can attend to both security and usability without sacrificing one for the other, or sacrificing one or both for innovation of functionality. As such, IRIS and CAIRIS are exemplars for design frameworks and software tools for secure and usable design, not the final word.

## Assumed Background

This book assumes you have some background or general interest in Security Engineering, Usability or Requirements Engineering, and are motivated to learn more about the other two. You are not expected to be an expert in any of these areas.

Those wanting a general overview of the field of Security Engineering will find *Security Engineering: A Guide to Building Dependable Distributed Systems* by Ross Anderson (2nd Edition, Wiley, 2008) useful. Because Threat Modelling techniques are useful when designing for security, readers may also be interested in *Threat Modeling: Designing for Security* by Adam Shostack (Wiley, 2014), as well as *Securing Systems: Applied Security Architecture and Threat Models* by Brook Schoenfield (CRC Press, 2015).

Many options exist for those wanting an overview of the field of Usability and related design topics; *Interaction Design: Beyond Human-Computer Interaction* by Jenny Preece, Helen Sharp and Yvonne Rogers (4th Edition, Wiley, 2015) provides both good coverage of the field, together with practical advice on the techniques covered in the book.

There are fewer options available when it comes to Requirements Engineering textbooks, but—given the attention IRIS and CAIRIS pays to KAOS and modelling



in general—then *Requirements Engineering: From System Goals to UML Models to Software Specifications* by Axel van Lamsweerde (Wiley, 2009) provides a useful overview of what Requirements Engineering can offer to design. Those desiring more hands-on advice on the art of eliciting, specifying and validating requirements would benefit greatly from *Mastering the Requirements Process: Getting Requirements Right* by Suzanne and James Robertson (3rd Edition, Addison-Wesley, 2012); this book is also the authority on the Volere framework, which informed how IRIS and CAIRIS deal with requirements.

When devising an IRIS process, it is also assumed you have some knowledge or experience on the design techniques that form the basis of the process. To illustrate the adaptation of usability design techniques, this book covers the persona technique in some depth; other techniques are given a lighter treatment. References are, however, provided at the end of each chapter for readers interested in using the framework techniques discussed in Sect. 4.3.

## How to Use This Book

This book was written to be read from cover-to-cover, but you can still skip to certain chapters based on your background and interests.

## Practitioners

Chapter 1 will help you put this book in context, and provides a roadmap for where to read next. If your focus is usability and incorporating interaction design for security, then Chap. 4 provides guidance on getting up and running with IRIS processes, and Chap. 6—using personas and scenarios as examples—provides ideas about how design techniques can be adapted to provide assurance necessary to support security decision-making.

If your background is in security, and you want ideas for incorporating Human Factors into your work, looking at how CAIRIS provides tool support for risk and task analysis in Chap. 5 will provide some inspiration, together with Chap. 9, which shows how security and usability can be incorporated into later stages of software design. You will also find that Chap. 11 provides practical advice on tackling security problems by treating solutions as innovations.

Irrespective of your background, you will find the case study chapters (Chaps. 7, 8 and 10) useful for seeing how security and usability processes and tools work in practice. If, while reading this book, you need more background material on why certain concepts are related in IRIS and CAIRIS, then Chap. 3 will provide a useful aide-memoire.

## Researchers

For those wishing to leverage or extend IRIS and CAIRIS, Part I will put whatever knowledge you have in context with complementary areas of design. The case study chapters will also put IRIS and CAIRIS in context; they illustrate the use of IRIS and CAIRIS, and present lessons learned as directions for future research.

The final chapters of this book describe numerous opportunities for future work. For example, Chap. 11 will benefit researchers in Security Economics by illustrating how economics and entrepreneurship can benefit each other. For researchers interested in the future potential of tool support for usable and secure design, then Chap. 12 provides examples of how the CAIRIS platform facilitates research and design in areas such as system-of-system engineering, practice integration with DevOps and the provision of rich exemplars for evaluating research as well as design.

## Educators and Students

This book was also written to support advanced undergraduate or postgraduate teaching in HCI-Security. The material in this book has already been used at Bournemouth University in a one-semester (10 weeks) course on *Security by Design* taken by final year students on the Forensic Computing and Security undergraduate degree programme, and postgraduate students on the Cyber Security and Human Factors postgraduate degree programme. The Indicative Learning Outcomes (ILOs) for that course are

1. Select and critically evaluate the usability criteria that security mechanisms have to meet to be usable in their contexts of use.
2. Apply techniques from interaction design and security engineering to design and evaluate secure systems.
3. Devise and evaluate supplemental measures that an organisation needs to ensure long-term, productive security.

The material used to satisfy ILOs 1 and 3 is heavily inspired by a course on *People and Security* designed by Angela Sasse that I once had the opportunity to deliver to students at University College London and the University of Oxford; this material is drawn from the academic literature in HCI-Security, Information Security and Economics of Information Security.

This book supports the satisfaction of ILO 2; Table 1 summarises the contribution this book makes to cogent lectures associated with this learning outcome. The seminars for these lectures use an on-going case study example based on *ACME Water*: a ‘human-centred’ specification exemplar that represents the operating environment of a hypothetical UK water company, and was designed on the basis of experiences gleaned carrying out the case study of Chap. 8.

**Table 1** Material from this book used to support a course on *Security by Design*

Lecture	Material used
Introduction	Challenges designing usable and secure software in Chap. 1 are used to motivate the course
Design techniques and tools	Material in Chap. 6 on personas and misusability cases provide examples of adapting interaction design techniques for security
Requirements	Examples of Security Requirements Engineering techniques and frameworks from Chap. 2, together with the IRIS framework in Chap. 4. The IRIS meta-model in Chap. 3 illustrates the relationship between Security, Usability and Requirements Engineering concepts
User Interfaces	Processes and tools for risk and task analysis with CAIRIS in Chap. 5
Architecture	Patterns and processes associated with Architectural Risk Analysis in Chap. 9, and the end-to-end webinos case study in Chap. 10
Security economics and entrepreneurship	Chapter 11 is the reading for the Security Entrepreneurship component of this lecture

## Using and Contributing to CAIRIS

This book illustrates the capabilities of CAIRIS, but does not go into the installation, setup and general use of the tool in any depth. You can, however, find full instructions on how to setup and use the platform on the CAIRIS website: <https://cairis.org>.

CAIRIS is an open-source software product. The source code has been released under an Apache License, and is freely available on GitHub. Raising an issue on the CAIRIS GitHub site is strongly encouraged if you find any problems with the software or its documentation. As an open-source project, CAIRIS is sustained by its community of developers and users. There are many ways interested readers can make a contribution to CAIRIS, from proposing new enhancements, raising bug reports or even contributing pull requests for any typographic errors or inconsistencies found in the available documentation. No contribution to CAIRIS is too small.

## Acknowledgements

The doctoral research that led to initial versions of both IRIS and CAIRIS was funded by EPSRC CASE Studentship R07437/CN001. I am very grateful to EPSRC and QinetiQ Ltd. for the sponsorship of this work. I am also grateful to the Department of Computer Science at the University of Oxford and Wolfson College for additional sponsorship of fieldwork and conference-related activities while

presenting this early work. Thanks go to Annamaria Carusi, Marina Jirotko, Angela Sasse and Andrew Simpson for their detailed and insightful comments on this work during its evolution from doctoral studentship proposal to submitted and corrected DPhil thesis. I would like to give special thanks to Dr. Ivan Fléchaïs, for his advice and encouragement, not only while carrying out the initial work leading to IRIS and CAIRIS, but follow-on work that took place at both Oxford and Bournemouth.

I am grateful to everyone on the DSS project, as well the water company behind the ACME name for allowing themselves to be used as case studies for evaluating IRIS and CAIRIS, finding the time to support my fieldwork, and provide feedback on the deliverables resulting from the respective IRIS processes applied. I am also grateful to everyone on the EU FP7 webinos project (FP7-ICT-2009-5 257103) for their support. The webinos project provided an important case study in the evolution of IRIS and CAIRIS; it provided a useful testbed for CAIRIS, and led to numerous improvements to the platform. The use of IRIS and CAIRIS to support the end-to-end design and development of webinos over its 3-year life also provided one of the first published long-term case studies for security and usability design.

The work described on Security Entrepreneurship in Chap. 11 was carried out while participating in the EPSRC sponsored *Science Innovation Programme* at the University of Oxford. The aim of the programme was to get research ideas out of the lab and into the next generation of technology innovation. However, after noticing the analogies between how technology entrepreneurs and security practitioners solve problems, I felt it necessary to get ideas on technology innovation out of practice and into the next generation of security research instead! I am grateful to Marc Ventresca and Victor Seidel at the Entrepreneurship Centre in the Säid Business School for their comments and insights on this work.

The on-going evolution of CAIRIS has benefitted from the generous assistance of the UK Defence Science and Technology Laboratory (Dstl). They provided invaluable feedback and comments on research resulting from the EPSRC-funded *Evaluating the Usability, Security, and Trustworthiness of Ad-hoc Collaborative Environments (EUSTACE)* project, which led to the work on trust modelling and trust expectations described in Sect. 12.1. More recently, they have supported on-going work on the use of CAIRIS to support risk assessment for systems of systems, which is briefly discussed in Sect. 12.1.

This book would not have been possible without the support of Bournemouth University (BU). BU has been generous in providing the time and resources needed to make this book a reality. Their support has ranged from pump-priming grants to kick start the work on CAIRIS shortly after I joined BU, matched funding for Ph.D. projects, support for research software engineers to support the development and maintenance of CAIRIS and general sympathy for the demands of writing a book while simultaneously managing multiple research and teaching commitments. My students at BU have also played an important role in this book by using IRIS and CAIRIS in exercises, assignments and, in several cases, undergraduate dissertations. Where things did not always go according to plan, they provided useful feedback and raised GitHub issues and, in several cases, pushed fixes to improve the stability and performance of the platform.

I would like to thank everyone who reviewed early drafts of this book for their suggestions for improving the presentation of this book, and smoothing away many rough edges: Andrea Atzeni, Benjamin Aziz, Huseyin Dogan, Michael Eaton, Jane Henriksen-Bulmer, Duncan Ki-Aries, Nancy Mead, Simon Parkin, Adam Shostack, Jacqui Taylor and Eylem Thron. I am particularly grateful to Andrew Simpson, whose comments and attention to detail went well beyond the call of duty.

On a personal note, I would like to thank my parents for their patience, support and love. Last, but by no means least, I would like to thank Claudia for her love, and for always being there for me.

Poole, UK  
February 2018

Shamal Failly

# Contents

## Part I Foundations

<b>1</b>	<b>Why Designing for Usability and Security is Hard</b>	3
1.1	Empowering the System Builders	3
1.2	Ubiquitous Technology	4
1.3	Integrating Processes	4
1.4	Growing Interests in Usable Security	5
1.5	IRIS and CAIRIS as Exemplars for Usability, Security, and Requirements Engineering Process and Tool Integration	5
1.6	Book Structure	6
1.6.1	Part 1: Foundations	6
1.6.2	Part 2: IRIS and CAIRIS	6
1.6.3	Part 3: Beyond Requirements	7
	References	8
<b>2</b>	<b>Usable and Secure Software Design: The State-of-the-Art</b>	9
2.1	Towards Usable Security Design	9
2.1.1	Themes in Information Security Design	9
2.1.2	Usable Security	11
2.1.3	A Case for Better HCI Integration?	14
2.2	Usability Design	15
2.2.1	HCI and Usability	15
2.2.2	User-Centered Approaches	17
2.2.3	Human-Centered Software Engineering	22
2.3	Specifying Security	23
2.3.1	Problem Frames	24
2.3.2	Goal-Oriented Approaches	27
2.3.3	Use Cases	35
2.3.4	Other Related Security and Usability Approaches	37

- 2.4 Specification Frameworks . . . . . 39
  - 2.4.1 RESCUE . . . . . 40
  - 2.4.2 SQUARE . . . . . 41
- 2.5 Tool-Support . . . . . 42
  - 2.5.1 Usability Design Tools . . . . . 42
  - 2.5.2 Security Requirements Engineering Tools . . . . . 43
  - 2.5.3 Visualising Design . . . . . 44
- 2.6 Summary . . . . . 45
- References . . . . . 46
- 3 A Conceptual Model for Usable Secure Requirements Engineering . . . . . 55**
  - 3.1 Introducing NeuroGrid . . . . . 55
  - 3.2 Overview . . . . . 56
  - 3.3 Environment Meta-Model . . . . . 57
  - 3.4 Asset Meta-Model . . . . . 59
  - 3.5 Task Meta-Model . . . . . 60
  - 3.6 Goal Meta-Model . . . . . 63
  - 3.7 Risk Meta-Model . . . . . 66
  - 3.8 Responsibility Meta-Model . . . . . 69
  - 3.9 Summary . . . . . 70
  - References . . . . . 71
- Part II IRIS and CAIRIS**
- 4 The IRIS Framework . . . . . 75**
  - 4.1 Perspectives . . . . . 75
  - 4.2 Converging Concepts . . . . . 77
  - 4.3 Framework Techniques . . . . . 77
    - 4.3.1 Grounded Theory . . . . . 79
    - 4.3.2 Personas . . . . . 80
    - 4.3.3 Activity Scenarios . . . . . 80
    - 4.3.4 Rich Pictures . . . . . 81
    - 4.3.5 KAOS . . . . . 82
    - 4.3.6 Volere Requirements Specification . . . . . 83
    - 4.3.7 Use Cases . . . . . 84
    - 4.3.8 AEGIS . . . . . 85
    - 4.3.9 Misuse Cases . . . . . 86
  - 4.4 Summary . . . . . 86
  - References . . . . . 87

- 5 Introducing CAIRIS: Tool-Support for Designing Usable and Secure Systems** . . . . . 89
  - 5.1 CAIRIS Design Principles . . . . . 89
    - 5.1.1 Familiarity . . . . . 89
    - 5.1.2 Extensibility . . . . . 90
    - 5.1.3 Process Centricity . . . . . 90
    - 5.1.4 Security and Usability Centricity . . . . . 90
  - 5.2 Tool Development Process . . . . . 91
    - 5.2.1 Initial CAIRIS Prototypes . . . . . 91
    - 5.2.2 From Desktop Tool to Software Platform . . . . . 91
  - 5.3 Architectural Design . . . . . 92
    - 5.3.1 Visual Interface Design . . . . . 92
    - 5.3.2 High Level Architecture . . . . . 94
    - 5.3.3 Physical Deployment . . . . . 96
    - 5.3.4 CAIRIS APIs . . . . . 97
  - 5.4 Tool-Support Characteristics . . . . . 98
    - 5.4.1 Automated Analysis . . . . . 98
    - 5.4.2 Model Visualisation . . . . . 105
    - 5.4.3 Model Reusability . . . . . 113
    - 5.4.4 Model Externalisation . . . . . 114
  - 5.5 Summary . . . . . 116
  - References . . . . . 117
  
- 6 Adapting Personas and Scenarios for Security and Usability Design** . . . . . 119
  - 6.1 Building Personas . . . . . 119
    - 6.1.1 Step 1: Identify Factoids from Source Data . . . . . 120
    - 6.1.2 Step 2: Affinity Diagram Factoids to Identify Behavioural Clusters . . . . . 120
    - 6.1.3 Step 3 (Optional): Affinity Diagram Behavioural Cluster Categories . . . . . 121
    - 6.1.4 Step 4: Categorise Behavioural Clusters by Behavioural Variables . . . . . 122
    - 6.1.5 Step 5: Write Persona Narrative . . . . . 122
    - 6.1.6 Persona Validation . . . . . 122
  - 6.2 Assumption Persona Argumentation . . . . . 123
    - 6.2.1 Toulmin’s Model of Argumentation . . . . . 123
    - 6.2.2 Developing Assumption Personas . . . . . 124
    - 6.2.3 Applying and Refining the Assumption Personas . . . . . 124
  - 6.3 Persona Cases . . . . . 126
    - 6.3.1 Building Persona Cases . . . . . 127
    - 6.3.2 Illustrated Example . . . . . 128
    - 6.3.3 Personifying Qualitative Models . . . . . 130



- 6.4 Supporting the Persona Creation Workflow with CAIRIS . . . . . 131
- 6.5 Misusability Cases . . . . . 131
  - 6.5.1 Eliciting Misusability Cases . . . . . 133
  - 6.5.2 Applying Misusability Cases . . . . . 134
- 6.6 Summary . . . . . 135
- References . . . . . 135
- 7 Case Study: Securing a Medical Data Portal . . . . . 137**
  - 7.1 About the Data Support Service (DSS) . . . . . 137
    - 7.1.1 Beginning the Project . . . . . 138
    - 7.1.2 e-Science and Security . . . . . 138
    - 7.1.3 The DSS User Community . . . . . 139
    - 7.1.4 Stakeholder Access . . . . . 139
  - 7.2 The IRIS Process . . . . . 140
    - 7.2.1 Persona Development and Scoping . . . . . 140
    - 7.2.2 Design Sessions . . . . . 141
    - 7.2.3 Misusability Case Analysis . . . . . 141
  - 7.3 Applying IRIS and CAIRIS . . . . . 142
    - 7.3.1 Persona Development . . . . . 142
    - 7.3.2 Design Sessions . . . . . 143
    - 7.3.3 Misusability Case Analysis . . . . . 143
  - 7.4 Lessons Learned . . . . . 148
    - 7.4.1 Persona Development . . . . . 148
    - 7.4.2 Design Sessions . . . . . 149
    - 7.4.3 Misusability Case Analysis . . . . . 151
  - 7.5 Summary . . . . . 153
  - References . . . . . 153
- 8 Case Study: Defending Critical Infrastructure Against Stuxnet . . . . . 155**
  - 8.1 About ACME Water . . . . . 155
    - 8.1.1 Instrumentation and Instrument Technicians . . . . . 156
    - 8.1.2 Legacy Equipment . . . . . 156
    - 8.1.3 Software Ubiquity . . . . . 157
    - 8.1.4 Information Security . . . . . 157
    - 8.1.5 Stuxnet . . . . . 158
    - 8.1.6 Plant Operations . . . . . 158
    - 8.1.7 The Enterprise SCADA Project . . . . . 159
  - 8.2 The IRIS Process . . . . . 159
    - 8.2.1 Scoping . . . . . 160
    - 8.2.2 Fieldwork . . . . . 160
    - 8.2.3 Usability and Security Analysis . . . . . 160
    - 8.2.4 Risk and Requirements Review . . . . . 162
  - 8.3 Applying IRIS and CAIRIS . . . . . 162

- 8.3.1 Scoping . . . . . 162
- 8.3.2 Fieldwork . . . . . 163
- 8.3.3 Usability and Security Analysis . . . . . 165
- 8.3.4 Risk and Requirements Review . . . . . 170
- 8.4 Lessons Learned . . . . . 171
  - 8.4.1 Scoping . . . . . 171
  - 8.4.2 Fieldwork . . . . . 172
  - 8.4.3 Usability and Security Analysis . . . . . 173
  - 8.4.4 Risk and Requirements Review . . . . . 174
  - 8.4.5 Qualitative Model Re-usability . . . . . 174
  - 8.4.6 Incorporating Stakeholder Activities  
into the Study Diagnosis . . . . . 175
- 8.5 Summary . . . . . 175
- References . . . . . 175

**Part III Beyond Requirements**

- 9 Analysing and Managing Architectural Risk . . . . . 179**
  - 9.1 Extending IRIS and CAIRIS for Architectural Design . . . . . 179
  - 9.2 Approach . . . . . 180
  - 9.3 Architectural Patterns . . . . . 180
  - 9.4 Attack Surface Metrics . . . . . 182
  - 9.5 Contextualised Attack Patterns . . . . . 182
  - 9.6 Architectural Risk Analysis . . . . . 185
    - 9.6.1 Architectural Pattern Specification . . . . . 185
    - 9.6.2 Attack Resistance Analysis . . . . . 185
    - 9.6.3 Ambiguity Analysis . . . . . 186
    - 9.6.4 Weakness Analysis . . . . . 186
  - 9.7 Example: An Architectural Risk Analysis of WebDAV  
for NeuroGrid . . . . . 187
    - 9.7.1 Architectural Pattern Specification . . . . . 187
    - 9.7.2 WebDAV Attack Surface Metric . . . . . 188
    - 9.7.3 Attack Resistance Analysis . . . . . 190
    - 9.7.4 Ambiguity Analysis . . . . . 192
    - 9.7.5 Weakness Analysis . . . . . 194
  - 9.8 Summary . . . . . 196
  - References . . . . . 196
- 10 Case Study: Securing An Internet of Things Middleware . . . . . 197**
  - 10.1 Learning from Research and Development Projects . . . . . 197
  - 10.2 Lessons Learned from E-Science and NeuroGrid . . . . . 198
  - 10.3 About Webinos . . . . . 199
    - 10.3.1 The Design and Development Process . . . . . 200
    - 10.3.2 The Usable Security Design Team . . . . . 201

10.4	Building Security into Webinos . . . . .	201
10.4.1	Context of Use Analysis . . . . .	201
10.4.2	Supporting Initial Requirements Elicitation and Specification . . . . .	204
10.4.3	Participatory Risk Analysis . . . . .	205
10.4.4	Analysing the Webinos Software Architecture . . . . .	206
10.4.5	Supporting Platform and Application Development . . . . .	207
10.4.6	Releasing Webinos Design Data . . . . .	208
10.5	Challenges . . . . .	209
10.5.1	User Research is Not Easy . . . . .	209
10.5.2	Usability is Not a Priority . . . . .	210
10.5.3	Technique Misappropriation is Easy . . . . .	211
10.5.4	Sustaining Adoption Through Implementation Requires Creativity . . . . .	211
10.6	Summary . . . . .	214
10.6.1	Learn to Work with Sub-optimal Data and Expertise . . . . .	214
10.6.2	Security Increases Sensitivity to Usability Problems . . . . .	214
10.6.3	Designing for Usability and Security Takes Time . . . . .	215
10.6.4	Design Research is a Key Element of Designing of Usability and Security . . . . .	215
	References . . . . .	216
<b>11</b>	<b>Evaluate Security as an Innovation . . . . .</b>	<b>217</b>
11.1	Security as an Innovation . . . . .	217
11.2	Innovation and System Building . . . . .	218
11.2.1	Creativity and Design . . . . .	218
11.2.2	Entrepreneurs as System Builders . . . . .	219
11.2.3	Models of Innovation . . . . .	220
11.2.4	Social Entrepreneurship Analogies . . . . .	221
11.3	Security Entrepreneurship and the Security Entrepreneur . . . . .	222
11.4	Security Entrepreneurship Techniques . . . . .	224
11.4.1	Security Chindōgu . . . . .	224
11.4.2	Innovation Value-Added Chain . . . . .	228
11.4.3	Social Network Analysis . . . . .	229
11.4.4	Security Premortems . . . . .	231
11.5	Towards Security Entrepreneurship . . . . .	233
11.5.1	Security Entrepreneurship Considered Harmful? . . . . .	234
11.5.2	How Should Security Entrepreneurship be Situated with Other Design Activities? . . . . .	234

- 11.5.3 How is Security Entrepreneurship Validated? . . . . . 235
- 11.5.4 Can Security Economics Help? . . . . . 236
- 11.6 Summary . . . . . 236
- References . . . . . 237
- 12 Further Applications of CAIRIS for Usable and Secure Software Design . . . . . 239**
  - 12.1 Environments as Stakeholder Perspectives in Systems of Systems . . . . . 239
  - 12.2 Threat Modelling . . . . . 240
  - 12.3 Trust Modelling . . . . . 242
  - 12.4 Realising “Design as Code”. . . . . 244
    - 12.4.1 Creating Assured Personas . . . . . 246
    - 12.4.2 Discovering Risks from Attackers and Threat Models . . . . . 248
    - 12.4.3 Contextualising the Implication of Security and Usability Design Decisions . . . . . 248
    - 12.4.4 Implications for CAIRIS . . . . . 251
  - 12.5 Human-Centred Specification Exemplars . . . . . 251
    - 12.5.1 Design Principles for Human-Centred Specification Exemplars . . . . . 252
    - 12.5.2 Evaluating Research with Human-Centred Specification Exemplars . . . . . 252
    - 12.5.3 Human-Centred Specification Exemplars and Archetypes . . . . . 253
  - 12.6 Summary . . . . . 253
  - References . . . . . 254
- Index . . . . . 255**

# List of Figures

Fig. 2.1	ISO 9241-11 usability framework . . . . .	16
Fig. 2.2	Human-centered design activities [52] . . . . .	18
Fig. 2.3	Example of a security requirement (modelled as an ellipse) constraining a context, modelled as a problem frame diagram [98]. . . . .	25
Fig. 2.4	KAOS goal model example . . . . .	28
Fig. 2.5	i* strategic dependency model example . . . . .	31
Fig. 2.6	i* strategic rationale model example . . . . .	32
Fig. 2.7	Use case model example . . . . .	35
Fig. 3.1	Architectural overview of NeuroGrid . . . . .	56
Fig. 3.2	Environment meta-model . . . . .	58
Fig. 3.3	Asset meta-model . . . . .	59
Fig. 3.4	Comparative asset values across environments . . . . .	60
Fig. 3.5	Task meta-model . . . . .	61
Fig. 3.6	Task meta-model example . . . . .	62
Fig. 3.7	Goal meta-model . . . . .	63
Fig. 3.8	Goal model example . . . . .	65
Fig. 3.9	Risk meta-model . . . . .	66
Fig. 3.10	Risk model example . . . . .	68
Fig. 3.11	Responsibility meta-model . . . . .	69
Fig. 3.12	Responsibility model example . . . . .	70
Fig. 4.1	IRIS perspective concepts . . . . .	76
Fig. 4.2	Mapping between Volere and IRIS requirements specification templates . . . . .	84
Fig. 5.1	CAIRIS screenshot . . . . .	92
Fig. 5.2	Textual view of CAIRIS model objects . . . . .	93
Fig. 5.3	Finding and updating for model elements . . . . .	94
Fig. 5.4	Graphical view of CAIRIS model objects . . . . .	94
Fig. 5.5	Component diagram of key CAIRIS components . . . . .	95
Fig. 5.6	Pipeline for visual model generation . . . . .	96
Fig. 5.7	Physical deployment diagram of CAIRIS . . . . .	97

Fig. 5.8 Asset model example . . . . . 106

Fig. 5.9 Task model example. . . . . 107

Fig. 5.10 Goal model example. . . . . 108

Fig. 5.11 Obstacle model example. . . . . 109

Fig. 5.12 Risk (top, red) and task usability (bottom, blue) colour charts . . . . . 109

Fig. 5.13 Risk analysis model before risk mitigation . . . . . 110

Fig. 5.14 Risk analysis model after risk mitigation . . . . . 112

Fig. 5.15 Responsibility model example . . . . . 113

Fig. 5.16 Importing a template OWASP threat into the current model. . . . . 114

Fig. 5.17 Threat and vulnerability directory XML schema. . . . . 115

Fig. 5.18 Project meta-data stored by CAIRIS. . . . . 116

Fig. 6.1 An affinity diagramming exercise . . . . . 121

Fig. 6.2 Conceptual model of assumption persona data . . . . . 125

Fig. 6.3 Toulmin model visualisation based on an individual characteristic. . . . . 125

Fig. 6.4 Contribution of Grounded Theory to personas . . . . . 129

Fig. 6.5 Misusability case with other design concepts . . . . . 132

Fig. 6.6 IRIS task meta-model updates for misusability cases . . . . . 133

Fig. 7.1 Instantiated process for the Data Directory specification. . . . . 140

Fig. 7.2 Characteristics and argumentation structure underpinning a misusability case . . . . . 145

Fig. 7.3 Misusability case contribution to goal model . . . . . 147

Fig. 8.1 IRIS process for the plant operations security policy . . . . . 161

Fig. 8.2 Finalised context diagram bounding the analysis for the security policy. . . . . 163

Fig. 8.3 Grounded theory model of plant operations staff security perception . . . . . 165

Fig. 8.4 Asset model associated with the security policy . . . . . 166

Fig. 8.5 Goal tree associated with exposed cabinets vulnerability . . . . . 169

Fig. 8.6 Penetration tester attacker profile . . . . . 170

Fig. 9.1 Architectural patterns meta-model. . . . . 181

Fig. 9.2 Contextualised attack patterns meta-model . . . . . 183

Fig. 9.3 Deployment diagram of NeuroGrid WebDAV architecture. . . . . 188

Fig. 9.4 Goals associated with WebDAV component. . . . . 189

Fig. 9.5 Component and Connector view of WebDAV architectural pattern. . . . . 190

Fig. 9.6 Assets associated with WebDAV server component . . . . . 191

Fig. 9.7 Obstacle model associated with Upload Malware attack pattern. . . . . 193

Fig. 10.1 Webinos development process . . . . . 200

Fig. 10.2 Context of use description concepts . . . . . 202

Fig. 10.3 Posters of persona characteristic, goals, and expectations. . . . . 205

Fig. 10.4 Use case map for “Local storage of credentials” use case. . . . . 207

Fig. 10.5 Consolidated concept map of requirements. . . . . 213

Fig. 11.1 The innovation ecosystem. . . . . 220

Fig. 11.2 Site authenticationware chindōgu . . . . . 226

Fig. 11.3 Ontology chart of chindōgu affordances . . . . . 227

Fig. 11.4 Innovation value-added chain for stakeholder groups . . . . . 229

Fig. 11.5 Existing security data social-network (left), and a  
social-network optimised for X.509 and XACML (right) . . . . . 230

Fig. 11.6 Misusability case argumentation model motivating  
a security premortem . . . . . 233

Fig. 12.1 DFD associated with broken instrument alarms during  
the day . . . . . 242

Fig. 12.2 A GRL model generated by CAIRIS within jUCMNav.  
The zoomed portion illustrates the impact of a strategy . . . . . 245

Fig. 12.3 Generated requirements specification document, which  
includes persona definitions and underpinning  
argumentation models. . . . . 247

Fig. 12.4 Enumeration threat modelled as an attack tree (left),  
and KAOS goal model integrating threat model  
into the network security requirements (right). . . . . 249

Fig. 12.5 Editing assets . . . . . 250

Fig. 12.6 Visualising a task’s security impact . . . . . 250

# List of Tables

Table 2.1	ISO 9241 usability definitions . . . . .	16
Table 3.1	IRIS meta-model core concepts . . . . .	57
Table 3.2	ISO 27001 definitions for confidentiality, integrity, and availability . . . . .	59
Table 4.1	Techniques overview . . . . .	78
Table 5.1	Principle to characteristic mapping . . . . .	98
Table 5.2	Task usability properties . . . . .	99
Table 5.3	Countermeasure task usability properties . . . . .	100
Table 5.4	IEC 61508 table for threat likelihood, vulnerability severity, and risk rating . . . . .	101
Table 5.5	IEC 61508 table for risk categorisation based on likelihood and severity . . . . .	101
Table 5.6	Project meta-data supported by CAIRIS . . . . .	116
Table 6.1	Behavioural variable types . . . . .	122
Table 6.2	Elements of Toulmin’s argumentation model . . . . .	123
Table 6.3	Characteristic argument example . . . . .	130
Table 7.1	Design session summary . . . . .	143
Table 7.2	Persona characteristics summary by behavioural variable type . . . . .	149
Table 7.3	Document and concept references associated with personas . . . . .	149
Table 7.4	IRIS concepts specified on completion of the DSS case study . . . . .	150
Table 8.1	In-situ interview summary . . . . .	164
Table 8.2	Policy goal examples . . . . .	165
Table 8.3	Tasks performed by plant operator persona . . . . .	168
Table 8.4	Vulnerabilities and affected assets . . . . .	168
Table 8.5	IRIS concepts specified on completion of the plant operations security policy case study . . . . .	172
Table 9.1	Damage-Effort Ratios . . . . .	182
Table 9.2	$DER_i$ input data . . . . .	191



Table 9.3	Upload Malware attack pattern . . . . .	192
Table 9.4	Leaf obstacles and mitigating requirements . . . . .	195
Table 10.1	Webinos architectural patterns . . . . .	206