Eliciting Usable Security Requirements with Misusability Cases

Shamal Faily, Ivan Fléchais
Department of Computer Science, University of Oxford
Email: {shamal.faily,ivan.flechais}@cs.ox.ac.uk

The Problem

? Current design techniques fail to engage developers in thinking about how their design decisions lead to both security and usability issues

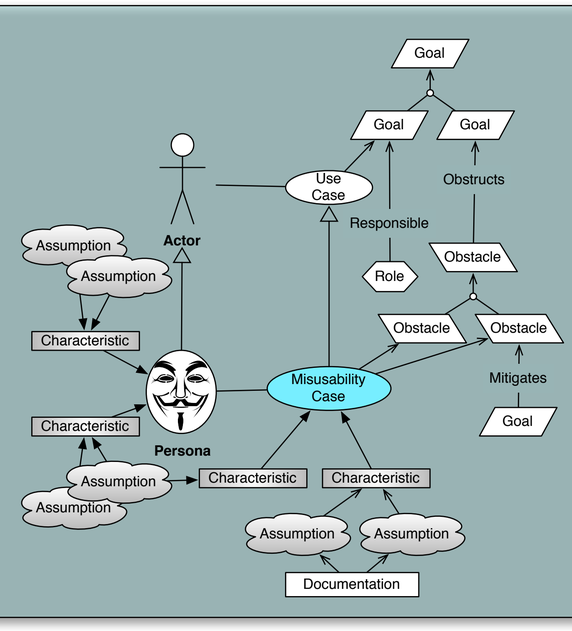
Our Approach

- ***Misusability***: design decisions leading to usability problems and system misuse
- Focus on unintentional, systemic effects
- Use design data to develop scenarios describing problems which lead to security misuse: ***Misusability Cases***
- Identify obstructions causing the Misusability Cases, and elicit goals to mitigate them

Situating Misusability Cases

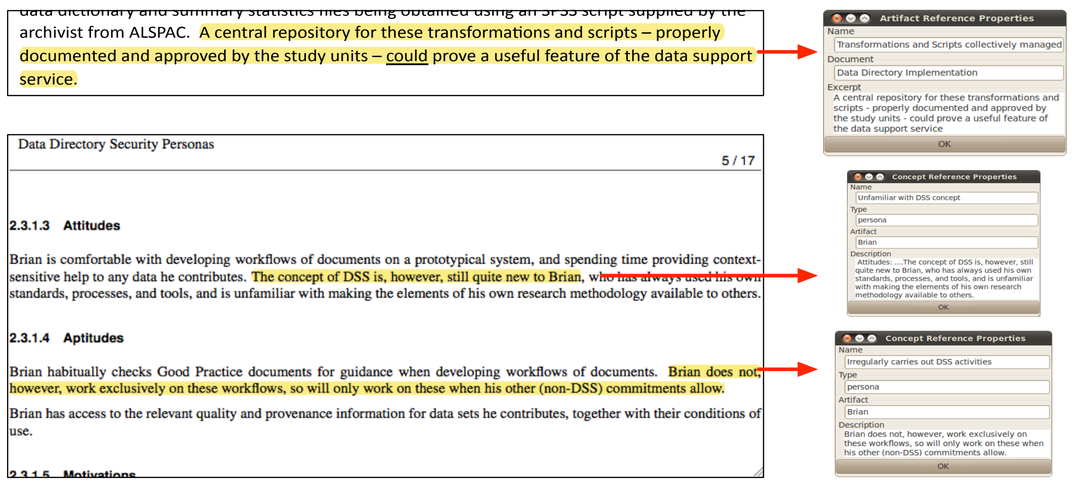
Misusability Cases do not exist in isolation, nor are they used during the early stages of requirements analysis. We assume goals have been elicited corresponding to the requirements a system needs to satisfy. We also assume that use cases [1] have been elicited describing episodes of system behaviour carried out by actors, and one or more personas [2] have been developed to contextualise these actors.

Misusability Cases are situated within the IRIS Meta-Model [3]. This meta-model illustrated how concepts from Requirements Engineering, Information Security, and HCI concepts can be integrated to support the elicitation and specification of secure system requirements.

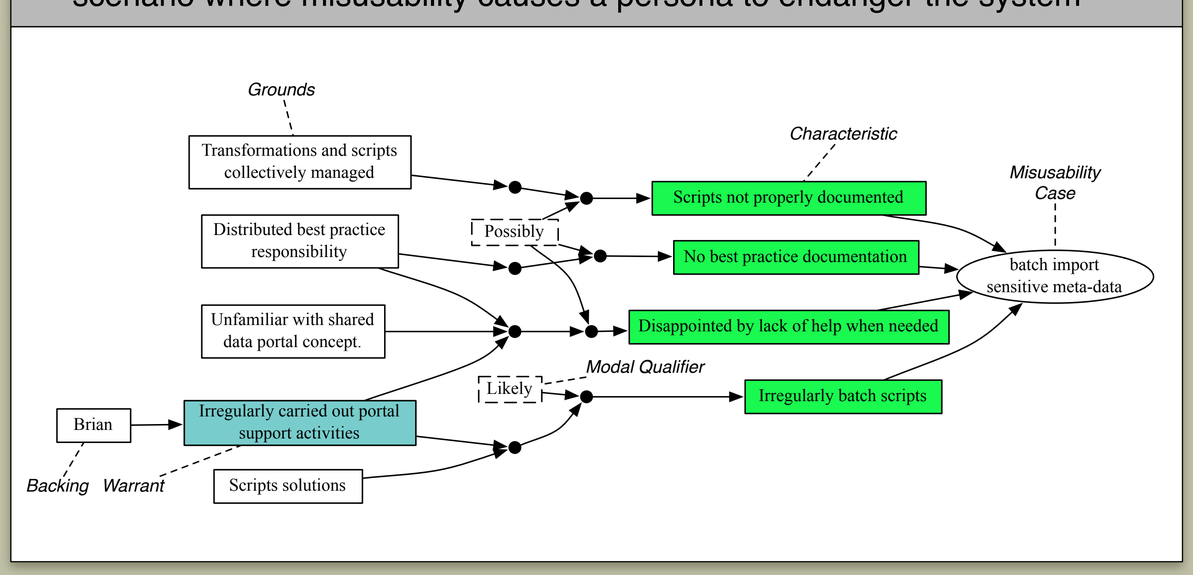


Eliciting and Applying Misusability Cases

1. Identify implicit assumptions from the design data giving rise to misusability



2. Using Toulmin's Model of Argumentation [4,5], model characteristics of scenario where misusability causes a persona to endanger the system



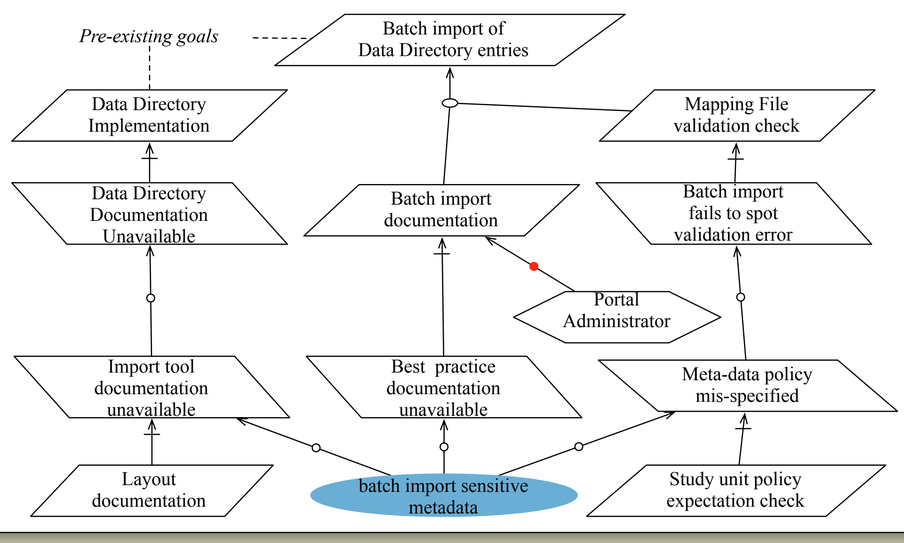
3. Write a Misusability Case based on the elicited argumentation model which operationalises one or more related use cases

Brian had spent most of the morning preparing data-sets ready for ingest into various sources. Some of the meta-data was for deep meta-data for local databases, while others were summarised meta-data targetted for the Data Directory. He hoped to use standards and guidelines on the gateway, but he was disappointed by the lack of anything useful that would help him. Never mind, Brian managed to organise his meta-data into the layout he managed to induce from some the XSLT scripts he downloaded.

After finally finishing the preparation of his data-sets, Brian created the mapping files needed for the data ingest process. Fortunately, most of them were very similar so most of the files he used were based on an initial template he created for one of his data-sets. Unfortunately, some of the policy setting were slightly different and, in the mapping file for the metadata for DSS, Brian inadvertently set a number of frequency metadata variables in the as publically accessible.

Brian entered a URI he had been provided for uploading meta-data to the Data Directory, and logged in using the data manager credentials. Brian then specified the mapping file corresponding to the meta-data he wanted to upload and hit the Upload button. Several minutes after clicking the Upload button, Brian received a message from the gateway saying the meta-data had been uploaded.

4. Using KAOS [6], elicit operationalising obstacles & identify mitigating goals



- Misusability Cases were used in a case study to help elicit security requirements for a portal for sharing medical study data.
- Goal models, system documentation, and related usability design artifacts were used as data sources for Misusability Case elicitation and specification.
- The CAIRIS Requirements Management tool [7] was updated to support the elicitation and visualisation of argumentation model elements.
- Of the 21 Obstacles and 6 key security requirements elicited, 15 obstacles and 4 requirements were elicited from Misusability Cases alone.

Future Work

✓ Misusability Cases are currently being applied to explore the impact of design ambiguity and user expectations about security and privacy on the EU FP7 webinos project.

References

[1] Cockburn, A., Writing Effective Use Cases. Addison-Wesley, 2001
[2] Pruitt, J., and Adlin, T. The Persona Lifecycle: Keeping People in Mind Throughout Product Design. Elsevier, 2006
[3] Faily, S., and Fléchais, I. A Meta-Model for Usable Secure Requirements Engineering. *ICSE Workshop on Software Engineering for Secure Systems* (2010), 126-135
[4] Toulmin, S., The uses of argument, Updated Edition. Cambridge University Press, 2003
[5] Faily, S., and Fléchais, I. The secret lives of assumptions: Developing and refining assumption personas for secure system design. *Proceedings of the 3rd Conference on Human-Centered Software Engineering* (2010),111–118
[6] van Lamsweerde, A. Requirements Engineering: from System Goals to UML Models to Software Specifications. Wiley, 2009
[7] Faily, S., and Fléchais, I. Towards tool-support for Usable Secure Requirements Engineering with CAIRIS. *International Journal of Secure Software Engineering* 1 (3), 2010, 56–70

Acknowledgements

This research was funded by the EPSRC CASE Studentship R07437/CN001, and by the EU FP7 *webinos* Project (FP7-ICT-2009-5 Objective 1.2) . We are also grateful to Qinetiq Ltd for their sponsorship of this work.