ONTOLOGY EDITING TOOL FOR INFORMATION SECURITY AND HUMAN FACTORS EXPERTS

John C. Mace, Simon Parkin and Aad van Moorsel
School of Computing Science & Centre for Cybercrime and Computing Security (CCCS)
Newcastle University, Newcastle*, U.K.*

Keywords: Information Security Ontology, Ontology Editor, Human Factors, OWL.

Abstract: Currently ontology development is facilitated by generic ontology editing tools which accommodate ontology experts, and not necessarily those individuals whose knowledge requires capture. Furthermore the process of knowledge capture is time consuming, error prone and requires appropriate technical skills. We propose a graphical editing tool for ontology development which simplifies knowledge capture. To demonstrate the tool's potential this paper explores the need for information security and human factors experts to capture their related knowledge in a dedicated ontology. Population of such an ontology would provide supporting information to enable organisations to make more well informed security policy decisions. Tailoring the tool for information security and human factors experts allows them to produce ontology content without being familiar with ontology construction and technologies. The tool is intuitive, requires no ontology component configuration, and provides mechanisms to guide users and reduce the potential for errors. Our tool allows information security domain experts to develop and extend the ontology, and organisations to tailor ontology content to their own requirements.

1 INTRODUCTION

There are instances where use of knowledge benefits from encoding the knowledge in an ontology. However, those in possession of the knowledge that is to be captured may be unable to familiarise themselves with ontology editing tools. This may be because they do not have the appropriate technical skills, or the time or inclination to understand ontology technologies.

Here we present an ontology editing tool that specifically serves a particular knowledge domain, in such a way that it negates the need for knowledge owners to understand ontology construction and ontology languages. Users construct a graphical representation of an ontology which can be automatically translated into an OWL (Web Ontology Language) encoded ontology.

The approach is demonstrated by applying the tool to serve information security practitioners. Within organisations it is essential to consider not only external security guidelines and standards (e.g. ISO27001 (BSI27001, 2005)), but also the impact of security upon people and processes within the organisation. If CISOs were able to augment their own expertise of IT security with knowledge of human-behavioural factors in information security they would potentially make more informed security policy decisions.

Applying this tool to information security policy-making demonstrates a solution where there is a need to formalise disparate, interconnected domain knowledge supplied by individuals who require assistance in the knowledge-capture process. It cannot be assumed that these individuals are experts in ontology development. We consider how Chief Information Security Officers (CISOs) and human factors experts would create and/or modify an information security ontology that could subsequently be tailored to specific organisations.

The tool addresses the need that these individuals have for a direct and intuitive way to populate an information security ontology with their interconnected knowledge of security issues and human-behavioural factors.

The rest of the paper is arranged as follows: Section 2 discusses the need for such an ontology and current construction approaches, alongside related work in ontology tools and construction.

Mace J., Parkin S. and van Moorsel A..
ONTOLOGY EDITING TOOL FOR INFORMATION SECURITY AND HUMAN FACTORS EXPERTS.
DOI: 10.5220/0003092302070212
In *Proceedings of the International Conference on Knowledge Management and Information Sharing* (KMIS-2010), pages 207-212
ISBN: 978-989-8425-30-0

Section 3 discusses the implementation of the editor tool. Concluding remarks are presented in Section 4.

2 BACKGROUND

2.1 Need for an Information Security Ontology

Currently, information security policy management decisions within an organisation are driven by external security standards and the experience of the Chief Information Security Officer (CISO) or similar.

However, policy decisions may affect or be influenced by the behaviour of individuals within the organisation. To make policy decision-making more effective, organisations (and more so CISOs) must have an awareness of the usability and other human-behavioural factors inherent in these decisions (Skidmore, 2003).

There is a need to organise and standardise disparate IT security policy-making information and human factors concerns in the form of a knowledge base or ontology. This would facilitate clear and effective communication within the IT-security management community, thereby further informing the security policy decision-making process.

2.2 Current Ontology Development

Currently the construction and/or modification of a security ontology is achieved through an ontology editing tool. These may come in the form of graphical or textual editors. Both types of editor allow content to be converted to a file written in an ontology language.

Both of these types of editor place similar demands on the user. The user must define the overall structure for potential ontology content, including concept types (ontology classes) and relationship types (ontology properties). As content is entered, each concept (ontology individuals) must be individually defined along with its class type, properties and relationship to other concepts.

This complex process assumes familiarity with ontology technologies. As such a CISO or Human Factors Researcher (HFR) may be unable to develop ontology content themselves, and would require either the assistance of an ontology expert or a dedicated ontology editing tool that hides ontology complexity.

2.3 Related Work

A large amount of work has been carried out in the field of security ontologies and with the rising interest in the Semantic Web this work is supplemented with a vast array of ontology creation tools.

The capture of security knowledge in an ontology has been shown to be viable through a number of studies e.g. (Parkin, 2009) and (Fenz, 2007). The work of Fenz et al (Fenz, 2007) incorporates the ISO27001 guideline (BSI27001, 2005) with a security ontology that considers the physical aspects of IT security management. This work allows organisations to audit security polices and assess whether they adhere to the ISO27001 guidelines.

For the successful development of a security ontology, the use of an ontology editing tool is required. A number of tools are already available e.g. Protege 3.4.4 (Stanford, 2010), OntoStudio (Ontoprise, 2010), TopBraid Composer (TopQuadrant, 2010) and NeOn Toolkit (NeOn, 2010). Protégé 3.4.4 offers form-based content entry, with ontology content presented in textual format. Ontology content is organised into class, property and individual hierarchies, in a manner whose level of complexity is appropriate for an ontology expert.

Another tool, SWOOP, (MINDSWAP, 2004) offers a Web browser style user interface aimed at the '*average Web user*' to facilitate the easy development and browsing of OWL ontologies. Ontology navigation and editing is carried out via a hyperlink based system. This approach enables both ontology and domain experts to contribute but is still reliant on ontology experts to contribute to the underlying formal structure for that knowledge.

Visualisation of an ontology during its construction or modification is of great advantage to the user and eases these processes immensely. There are a number of visual ontology creation tools using OWL as a base language e.g. GrOWL (Vermont, 2006), OWL-S Editor (Scicluna, 2004) and SemanticWorks (Altova, 2010) all of which illustrate the ontology in a UML format. SemTalk 2 (Semtation, 2005) uses Microsoft Visio's functionality to create and modify ontologies graphically, again in a UML format, translating ontologies automatically to an OWL ontology file.

Although ontology creation is aided by the graphical functions of these tools they still remain relatively complex, require a certain amount of initial training; and are generic in nature and not designed specifically for security ontology creation unlike our proposed tool.

The development of our tool has been based on looking at the positive and relevant features from currently available applications. The result is an ontology editing tool designed specifically for security domain experts, allowing them to capture their knowledge in an easy and intuitive way while removing the need to know of ontology construction techniques.

2.4 Requirements

After review of related work and the problem space, the main requirements of an ontology editing tool for CISOs and HFRs have been identified:

- **Encourage Knowledge Capture.** The tool must capture the unstructured knowledge of domain experts within a suitable formalised ontology structure.
- **Hide Technical Details.** A domain expert should only need to concern themselves with adding new information, without a need to manage the underlying ontology structure.
- **Assist Knowledge Owners.** Knowledge owners perceive their knowledge in their own way. To formalise knowledge correctly, users should be actively assisted by the editor tool during knowledge entry
- **Minimise Inconsistencies.** There must be mechanisms to minimise errors occurring in the knowledge capture process.

3 IMPLEMENTATION

3.1 Components

In Figure 1 the ontology editor's main components can be seen.

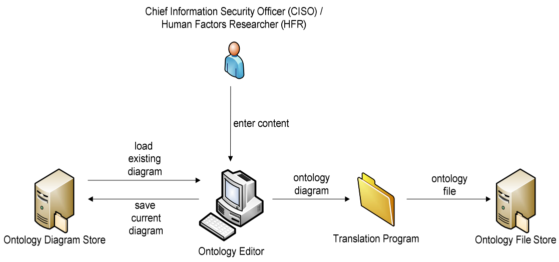


Figure 1: Overview of ontology editor's components.

3.1.1 Foundation Ontology

The editor uses an existing ontology (Parkin, 2009) to define security issues and human-behavioural factors, and crucially relate them to concepts within an organisation's security policies. For these purposes we use a modified version of the ontology.

The concepts represented in the ontology are shown in Figure 2. The ontology represents those information `Assets` that either must be secured or which are crucial to an information security management process. A `Vulnerability` may represent the security and usability weaknesses of an `Asset` that may promote or inhibit certain employee behaviours. A `Vulnerability` may be '`exploitedBy`' a `Threat` which renders the `Asset` insecure or unusable (thereby also potentially affecting productivity). Exploitation of a `Vulnerability` may be intentional or accidental.

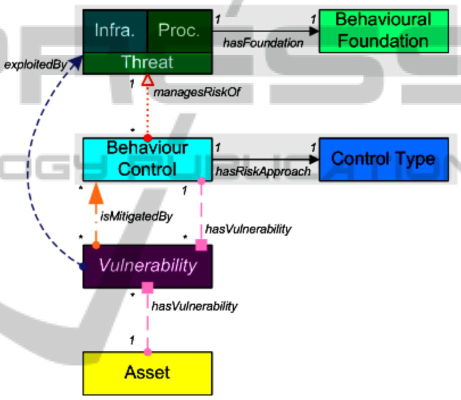


Figure 2: Overview of the information security and human factors ontology.

A `Threat` may be either an `Infrastructure Threat` or a `Procedural Threat`. The former represent activities that directly affect security mechanisms, whereas the latter represent security events that impact upon an individual and their behaviour. The `Behavioural Foundation` of a `Procedural Threat` classifies behaviours to indicate the concerns that they raise within an organisation (e.g., a person's memory capabilities or attitude towards security).

A `Vulnerability` may be '`mitigatedBy`' a `Behaviour Control`, which represents a procedural activity that a CISO can enact to manage interactions between humans and organisational security controls. The associated `Risk Control Type` indicates a risk management approach, such that a `Behaviour Control` ‘`managesRiskOf`’ a `Threat`.

3.1.2 Ontology Editor

The ontology editor offers the user a simple graphical interface where they can enter and capture their knowledge in graphical form (i.e. a graphical representation of an information security ontology).

All aspects of the security ontology structure (e.g. classes and properties) are pre-defined, and with the integrated help system, diagram construction (i.e. ontology population) is intended to be simplified and intuitive.

The underlying ontology is not presented to the user but used implicitly to derive the graphical elements and to translate the diagram into OWL.

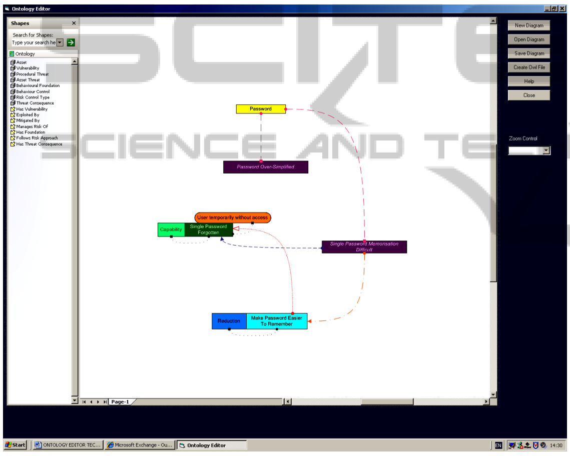


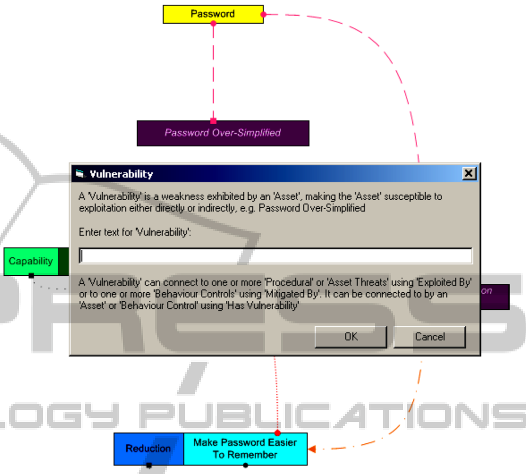
Figure 3: Screenshot of editor’s user interface.

To the right of the user interface are the available controls (new/open/save diagram, create OWL file, etc).

The main window of the editor interface, as shown in Figure 3, is where ontology diagram construction takes place, using an embedded instance of the Microsoft Visio 2007’s drawing control (Microsoft, 2010). Use of Visio is appropriate for our tool as it is intended for use in large organisations that are assumed to already have Microsoft Office software available.

On the left of the drawing control is a list of pre-defined shapes available for constructing diagrams. Each shape contains individual data (name, colour, size, etc) and can be re-used by dragging and dropping onto the drawing page. The shapes consist of boxes (ontology concepts) and arrows (ontology relationships), each represented in a different colour to help the user differentiate between them.

When using arrows to form ontology relationships, the Visio auto-connect feature is applied to assist in the process. This same feature also maintains connections between shapes when a shape is moved around the drawing area. This then allows the user to manipulate the diagram according to their needs without needing to manage connections.



A 'Vulnerability' is a weakness exhibited by an 'Asset', making the 'Asset' susceptible to exploitation either directly or indirectly, e.g. Password Over-Simplified

Enter text for 'Vulnerability':

A 'Vulnerability' can connect to one or more 'Procedural' or 'Asset Threats' using 'Exploited By' or to one or more 'Behaviour Controls' using 'Mitigated By'. It can be connected to by an 'Asset' or 'Behaviour Control' using 'Has Vulnerability'

Figure 4: Dialog box for adding new concept.

A number of mechanisms are used to restrict the potential for errors in the knowledge capture process. Only certain boxes can be associated with each other using certain types of arrows, removing the potential for invalid connections. A further error handling feature of the editor is the detection of unconnected shapes (isolated boxes, unconnected arrows, etc). If such “hanging” shapes are found, saving of the ontology file is halted until the user resolves the relevant errors.

An integrated help system is in place to aid the user in diagram construction. When a new box (concept) is created a dialog box explains what the box represents, how it is used and how it may be connected to other boxes in the diagram. An example is shown in Figure 4. “Tool tips” actively provide explanations of ontology components and editor functions when the user floats the mouse cursor over them.

3.1.3 Ontology File

When an ontology diagram is saved, information about the ontology content is stored in XML format. This facilitates translation into an Ontology File, wherein ontology content is translated into the Web Ontology Language (OWL) (W3C, 2004) format.

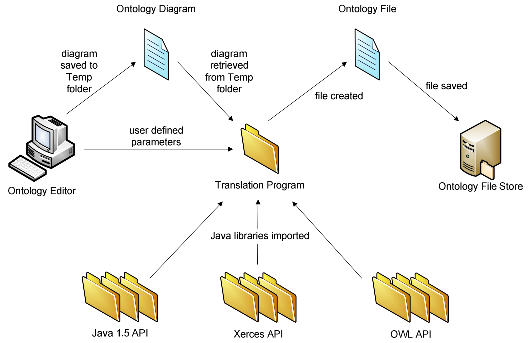


Figure 5: Overview of Translation Program's components.

OWL files can potentially be processed automatically by other software programs, thereby providing scope for expert knowledge to be used in various ways. Once created, all ontology files are stored in a user designated file store.

Further knowledge may be added to pre-existing ontology diagrams which is then propagated to the corresponding ontology file. This removes the need to regenerate the entire ontology.

3.1.4 Translation Program

The Translation Program processes Ontology Diagrams to create Ontology Files.

The Java Translator Program is written in the Java SE v1.5 programming language (Oracle, 2010) and is deployed as an executable Java archive on a user's machine.

Figure 5 provides an overview of the components involved in the translation process. This process is two-fold: first an Ontology Diagram file is processed to obtain relevant data; secondly, that data is transformed into the OWL format before being compiled into an ontology file. An entire diagram can be translated into an ontology file via a single operation, removing the need for successive processing of each knowledge fragment.

Data is retrieved for each shape (instance) based on its type, contained text (entered by user) and any connections to other shapes. To write this data into an ontology file, libraries from the OWL Java API (Manchester, 2010) are used.

The classes and properties definitions of the underlying information security ontology are hard-coded into the Translation Program, and are written directly to the ontology file. By predefining the ontology structure the user is not expected to understand or define ontology language constructs.

4 CONCLUSIONS

Information security management knowledge and security human factors knowledge is currently vast and fractured. This knowledge must be collated to enable organisations to make more well informed security policy decisions. Ontologies serve as a means to store this knowledge.

Current ontology development utilises editing tools which are mostly aimed at ontology experts, and not necessarily those who hold the knowledge that must be captured. This paper shows that there is a need for an editing tool designed for information security and human factors experts (amongst others) to capture their interrelated knowledge in the form of an ontology.

We describe a tool that produces machine-readable OWL ontology files, derived from a diagrammatic representation of information security and human factors knowledge. Ontology concepts and relationships are automatically translated from diagram components into an OWL ontology file.

Our tool simplifies the ontology development process, requiring little or no knowledge of ontology construction on the part of the knowledge owner.

Future work hopes to see the development of a community Web-based version of the editing tool to facilitate collaborative knowledge capture across wider communities of information security and human factors experts.

ACKNOWLEDGEMENTS

The authors are supported in part by UK Technology Strategy Board (TSB), grant nr. P0007E ("Trust Economics") and HP Labs Innovation Research Program, award ID 2009-1052-1-A ("Prediction and Provenance for Multi-Objective Information Security Management").

REFERENCES

Altova, 2010. *SemanticWorks Semantic Web tool - Visual RDF and OWL editor*, http://www.altova.com/semanticworks.html, last visited 18/03/10

British Standards Institution (BSI UK), 2005. *BS ISO/IEC 27001:2005 - Information Technology - Security Techniques - Information Security Management Systems – Requirements*. BSI.

Fenz, S., Goluch, G. , Ekelhart, A. , Riedl, B., Weippl, E., 2007. *Information Security Fortification by*

Ontological Mapping of the ISO/IEC 270001 Standard. Proceedings of the 13th Pacific Rim International Symposium on Dependable Computing (PRDC2007). IEEE Computer Society.

Manchester, University of , 2010. *The OWL API*. http://owlapi.sourceforge.net/, last visited 18/03/10

Microsoft Corporation, “Visio Home Page – Microsoft Office Online”, 2010. http://office.microsoft.com/en-us/visio/default.aspx, last viewed 18/03/10

MINDSWAP Reasearch Group, 2004. SWOOP – A Hypermedia-based Featherweight OWL Ontology Editor. http://www.mindswap.org/2004/SWOOP/, last visited 22/07/10

NeOn, 2010. *NeOn toolkit*. http://neon-toolkit.org/wiki/Main_Page, last visited 22/07/10

Ontoprise GmbH, 2010. *OntoStudio*. http://www.ontoprise.de/en/home/products/ontostudio/, last visited 22/07/10

Oracle Corporation, 2010. *Java SE Overview - at a Glance*. http://java.sun.com/javase/, last visited 18/03/10

Parkin, S. E., van Moorsel, A., Coles. R., 2009. *An Information Security Ontology Incorporating Human-Behavioural Implications*, Proceedings of the 2nd International Conference on Security of Information and Networks (SIN).

Scicluna, J., Abela, C., Montebello, M., 2004. *Visual Modeling of OWL-S Services*. Computer Science Annual Research Workshop (CSAW) ’04.

Semtation GmbH, 2005. *SemTalk – Ontology Editor for MS Office*. http://www.semtalk.com/fr_index.html?/Ontologien.htm, last visited 18/03/10

Skidmore, P. “Beyond Measure”, Demos, 2003

Stanford Center for Biomedical Informatics Research, 2010. *Welcome to Protégé.* http://protege.stanford.edu/, last viewed 18/03/10

TopQuadrant, 2010. *TopBraid Composer*, http://www.topquadrant.com/products/TB_Composer.html, last visited 23/07/10

Vermont, the University of, 2006. *Welcome to GrOWL Website*, http://www.uvm.edu/~skrivov/growl/, last viewed 18/03/10

W3C, 2004. *OWL Web Ontology Language Overview*. http://www.w3.org/TR/owl-features/, last viewed 18/03/10