The Design, Instantiation, and Usage of Information Security Measuring Ontology

Antti Evesti, Reijo Savola, Eila Ovaska, Jarkko Kuusijärvi
VTT Technical Research Centre of Finland
Oulu, Finland
e-mail: antti.evesti@vtt.fi, reijo.savola@vtt.fi, eila.ovaska@vtt.fi, jarkko.kuusijarvi@vtt.fi

***Abstract*—Measuring security is a complex task and requires a great deal of knowledge. Managing this knowledge and presenting it in a universal way is challenging. This paper describes the Information Security Measuring Ontology (ISMO) for measuring information security. The ontology combines existing measuring and security ontologies and instantiates it through example measures. The ontology provides a solid way to present security measures for software designers and adaptable applications. The software designer can utilise the ontology to provide an application with security measuring capability. Moreover, the adaptable application searches for measures from the ontology, in order to measure a security level in the current run-time situation. The case example illustrates the design and run-time usage of the ontology. The experiment proved that the ontology facilitates the software designer's work, when implementing security measures for applications that are able to retrieve measures from the ontology at run-time.**

Keywords-adaptation; run-time; quality; measure; security metric; software

I. INTRODUCTION

Software applications running on devices and systems may face needs for changes due to alterations happening in their execution environments or intended usages. These changes may have a considerable effect on the security requirements of the software system. Moreover, emerging security threats and vulnerabilities may affect the achieved security level. However, the software system is required to achieve a desired security level in these changing circumstances [1]. Therefore, the software has to be able to observe the security level at run-time, measure the fulfilment of the security requirements, and adapt itself accordingly. However, measuring the security level at run-time requires the correct measures and measurement techniques for each situation. Defining the measures and the measuring techniques is a time consuming task and requires the use of experts from different domains. Thus, it is important to present the defined measures in a universal and reusable form. In addition, problems concerning how to present these measures, the measuring techniques, and their mutual relationships have to be solved in a way that facilitates run-time security measuring. Ontologies provide a possibility to manage this knowledge, making it possible to describe different security requirements and ways to measure the fulfilment of these requirements.

Ontologies are utilised in [2] to achieve the required quality of the software at a design-time. Thus, it is reasonable to utilise ontologies as a knowledge base for quality management at the run-time. Furthermore, the work in [3] presents the architecture for developing software applications with security adaptation capabilities – the presented approach assumes that the knowledge required for security monitoring and adaptation is available from ontologies.

In this work, we will present a novel ontology for the run-time security measuring – called Information Security Measuring Ontology (ISMO). ISMO combines a terminology from a software measurement area in general and the security related terminology. In addition, a few security measurements are added to the ontology in order to instantiate it. The novelty of our work comes from this combination – based on our current knowledge, there isn't any other ontology which describe security measuring in a run-time applicable way. The content of ISMO can be enhanced after the software application has been delivered. Hence, the measuring process is based on the up-to-date specifications of security measures. The purpose of this new ontology is to make it possible for software applications to utilise security measures during run-time in changing environments. Therefore, it is possible for the application to measure the fulfilment of its security requirements and adapt the used security mechanisms if the required security is not met. In other words, the measuring acts as a trigger for the adaptation. However, to achieve software applications with a measuring capability, the developer must have implemented a set of measuring techniques as a part of it. Thus, ISMO also provides input for developers – presenting what measuring techniques are to be implemented and how.

The remainder of the paper is organised as follows: Section 2 provides background information; Section 3 presents an overview of the combined ontology and mentions some security measurements. Section 4 instantiates the defined ontology. Section 5 explains how the ontology is utilised. A case example is presented in Section 6. Finally, a discussion and conclusions close the paper.

 ISBN: 978-1-61208-130-4

II. BACKGROUND

ISO/IEC defines security in [4] as follows: "The capability of the software product to protect information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them." Furthermore, in some sources security is thought to be a composition of confidentiality, integrity and availability [5, 6]. In [7], these security sub-attributes are called security goals.

Zhou [8] defines ontology as a shared knowledge standard or knowledge model, defining primitive concepts, relations, rules and their instances, which comprise topic knowledge. It can be used for capturing, structuring and enlarging explicit and tacit topic knowledge across people, organizations, and computer and software systems.

Blanco et al. [9] lists several security ontologies in their work. In addition, our earlier work [10] also compares a number of security ontologies, particularly those that are applicable for run-time usage. It is noticed in [10] that security ontologies for run-time usage exist – especially for service discovery and matchmaking, for example, ontologies from Denker et al. [11] and Kim et al. [12]. In addition, security ontologies which concentrate on software design and implementation phases also exist, e.g., works from Savolainen et al. [13] and Tsoumas et al. [14]. From these ontologies, only the work from Savolainen et al. [13] takes measurements into account, by presenting a high level classification for different security measures. However, the most extensive information security ontology at the moment is the one proposed by Herzog et al. [7], called an ontology of information security – abbreviated as (OIS) in this work. The OIS is intended to provide a general vocabulary or an extensible dictionary of information security. It is applicable at design and run-time alike, and it contains more concepts than all the above mentioned security ontologies altogether. Thus, this ontology provides a sound starting point for defining the concepts of ISMO.

The OIS does not contain concepts for describing measures. Therefore, the Software Measurement Ontology (SMO) [15] is utilised for measurement definitions. The SMO presents the generic measurement terminology related to software measurements. In other words, ontology is quality attribute independent. The SMO collects and aligns terminology from several standards of software engineering, software quality metrics, and general metrology. It is important to notice that the SMO uses a term measure instead of metric. Thus, the measure term will be used in this work. The SMO divides measures into three sub-classes: namely a base measure, derived measure, and indicator – all of which inherit the same relationships from other concepts. The base measure is an independent 'raw' measure. A derived measure is a combination of other derived measures and / or base measures. Finally, an indicator can be a combination of all of these three types of measures. The complexity of these measures increases when moving from base measures to derived measures and further on to indicators. In literature, base measures and derived measures are also called direct and indirect measures; however we will follow the terminology defined by the SMO.

Hence, this work draws mappings between the OIS and SMO, instead of defining a new ontology from scratch. This is considered to be reasonable since a remarkable effort has been invested into these existing ontologies and both are scientifically reviewed and accepted. In addition, the reuse of existing ontologies is suggested in [16] as one potential approach for ontology development.

III. THE DESIGN OF INFORMATION SECURITY MEASURING ONTOLOGY

This section describes how the combined ontology ISMO is designed. SMO [15] contains 20 generic measurement related concepts and their relationships. Thus, security measures will be used to instantiate ontology for security measuring purposes – creating base measures, derived measures, and indicators. On the other hand, OIS [7] contains concepts related to threats, assets, countermeasures, security goals, and the relationships between those concepts. In addition, the OIS describes a couple of vulnerabilities and how these act as enablers for threats. The OIS already contains some of these concepts as an instantiated form, such as the security goals of authentication, integrity, etc.

The purpose of combining these two ontologies is to achieve an ontology that makes it possible to measure the fulfilment of security requirements, i.e., security goals and levels. In other words, the purpose is to enable an operational security correctness measurement, as called in [17]. Therefore, the requirements are described by a means of vocabulary from the OIS. The requirements fulfilment is measured with indicators – which combine several measures – defined in the SMO. The security measures, i.e., indicators, are different for each security goal, e.g., a level of authentication or non-repudiation is measured with different measures. However, these measures can utilise the same base measures. On the other hand, the same security goal can be achieved with different countermeasures, which in turn might require their own measures. For instance, the authentication level, which is achieved, is measured in a different way when a security token is used instead of a password authentication. Hence, there are only a few concept-to-concept mappings between these two ontologies, but additional mappings appear when the measures are instantiated. By using a terminology from ontologies, a mapping refers to the property between the concepts. Adding mappings for instantiated measures requires domain expertise, i.e., the capability to recognise applicable measuring techniques for a particular security goal and related mechanisms. Furthermore, the mapping requires a capability to recognise threats which affect to the particular security goal and/or mechanism. Mappings at the instantiation phase are described more detail in Section IV.

Fig. 1 shows an overview of the combining process. Firstly, the mappings between the concepts are drawn. Secondly, security measure instances are added and related mappings for each measure are defined. Thus, the SMO is used as a guideline as how to define the instances of security measures.

 ISBN: 978-1-61208-130-4

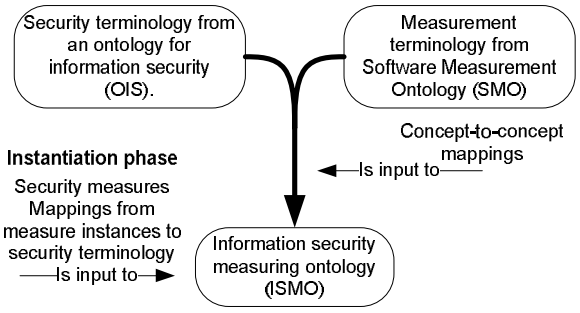

Figure 1. An overview of the combining process.

TABLE I shows mapping properties between concepts from the SMO and OIS. These are mappings made from concept to concept. Each *SecurityGoal* has an *Indicator* – intended for measuring the fulfilment of the goal. The SMO uses the term *QualityModel* for defining measurable concepts. *QualityModel* is a quality attribute dependent, i.e., security in this case. Thus, the *QualityModel* concept is related to *SecurityGoal*. The *MeasurableConcept* from the SMO is also mapped to the *SecurityGoal*, through the means of the *isDefinedFor* property.

In the SMO, *MeasurableConcept* relates to *Attribute*, meaning a characteristic that will be measured. Thus, *Attribute* can relate to countermeasures, threats, or assets, depending on the measure which is used. *hasMeasurableAttribute* is optional, meaning that mappings to the attributes are made during the phase when the measures are instantiated.

TABLE I. MAPPING PROPERTIES BETWEEN SMO AND OIS

Concept from OIS	Mapping property (direction)	Concept from SMO
SecurityGoal	hasIndicator (->)	Indicator
SecurityGoal	isRelatedTo (<-)	QualityModel
SecurityGoal	isDefinedFor (<-)	MeasurableConcept
Countermeasure, Threat, Asset	hasMeasurableAttribute (->) (optional)	Attribute

A. Security Measures

The overall security level of the product can be represented by a combination of relevant security attribute measures. However, it is not possible to cover all security measures in this work. Consequently, we will concentrate on user authentication, and thus, the ISMO is instantiated by these measures.

In [18] measures for various security goals (e.g., authentication, integrity, etc.) are defined by using a decomposition approach introduced by Wang et al. in [19]. Authentication can be decomposed into five components – called BMCs (Basic Measurable Components) – as follows: Authentication Identity Uniqueness (AIU), Authentication Identity Structure (AIS), Authentication Identity Integrity (AII), Authentication Mechanism Reliability (AMR), and Authentication Mechanism Integrity (AMI). Savola and Abie [18] define equations for these BMCs, and in addition, the equation for combining Authentication Strength (AS) from these BMCs. AS is an aggregated user-dependent measure that can be utilized in authentication and authorization.

The user-dependent AS results can be combined into a system-level AS, which can be utilized in run-time adaptive security decision-making [18]. When considering a software application that measures its security level at run-time, AIS, AII, and AMR are particularly applicable. In other words, an application cannot measure AIU and AMI, as the information which is required for these measures is only available at the server side where the application will be authenticated. Thus, in this work, the measures for the AIS will be used as example measures.

To measure AIS, we utilise a measure intended for situations where the authentication is based on a password – called the structure of the password. It is commonly understood that the structure of a password, i.e., the length and variation of the symbols, affects the achievable authentication strength. Therefore, we divide passwords into groups such as: i) a PIN code containing four numbers, ii) a password containing 5-9 lower case characters, and iii) a password containing over 10 ASCII symbols. Intuitively, *group i* provides the worst authentication level, *group ii* offers an increased authentication level, and *group iii* is the best alternative.

Another measure that we utilise for password based authentication is the age of the password, i.e., how long the same password has been used. This measure can be used for two different purposes. Firstly, to measure the security policy fulfilment, e.g., a policy can define that the password has to be changed every three months. Secondly, the measure can be utilised as a factor of measuring the authentication strength by utilising more complex analysis models. The age of the password is also mapped to the AIS from BMCs.

These two measures are simple to understand, and thus, provide a good starting point for instantiating ISMO. The graphs in Fig. 2 illustrate how the password strength is affected by the structure and age of the password. These are however merely examples and we do not claim that these affects are linear.

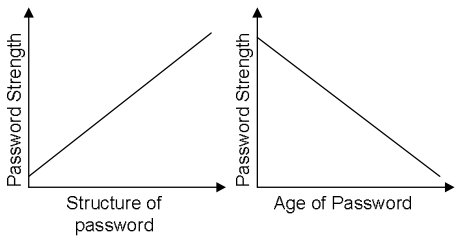

Figure 2. Conceptual correlation graphs for the authentication measures.

IV. THE INSTANTIATION OF INFORMATION SECURITY MEASURING ONTOLOGY

In this section, the authentication related measures are instantiated as a part of the ISMO and the required mappings are added. Fig. 3, Fig. 4, and Fig. 5 present the instantiated ontology – rectangles depict the concepts from SMO and ellipses refer to concepts from OIS. The name of each

concept is presented in the figures and separated from the instance name by a colon, i.e., ConceptName : InstanceName. The property mappings between these two ontologies are presented in bold fonts. For reasons of clarity, the instantiated ontology is presented in three separated figures. Consequently, some concepts may appear in each figure, but from a different viewpoint, i.e., presenting different properties.

The SMO contains the concept *MeasurableConcept*, which corresponds conceptually to the BMCs, which are defined in [18]. Thus, AIS BMC is instantiated in the ontology as a *MeasurableConcept*. The concept *QualityModel* refers to security goals in this work. Hence, there is a mapping property from the *QualityModel* concept to Security goals (authentication, confidentiality, etc.), as mentioned in Section 3.

The *PasswordAge* measure (Fig. 3) is the first measure instance which is added to the ISMO. In the SMO, measures are defined for attributes and these attributes are related to the measurable concept. AIS is an instance of the measurable concept and *PasswordAge* is one of the related attributes. This attribute is measured through the means of an instantiated derived measure, called *UsageTime*. Hence, the derived measure is not purely security related, and can also be applied for other attributes, e.g., the usage time of the CPU in performance measurements. The derived measure *UsageTime* is calculated with a measurement function – defining that *UsageTime* is the current date minus the starting date. The calculation of the value for this measurement function requires that a base measure instance called *Date* is used. *Date* is a base measure, meaning that it is not dependant on other measures and its value is measured by the measurement method. The measurement method for the *Date* measure is simple: taking the date value from the system clock. Defining *UsageTime* as a derived measure may seem like an overestimation. However, detailed definitions are required in order to achieve a measuring ontology that supports run-time security measuring.

The second measure – presented in Fig. 4 – is connected to the AIS via an attribute called *PasswordStructure*. The attribute is measured with an instantiated indicator called *PasswordType*. Indicators are calculated using an analysis model. In this context, the analysis model is a set of rules, which can be thought as if-then-else statements. We have decided to use statements which are very close to the natural English language, so that the analysis models could be updated without an extensive knowledge on programming. The statements of the analysis model can be updated later on to, e.g., the standard SPARQL [20] queries. The analysis model itself is saved as a string literal, so it can be easily changed into a SPARQL statement. For simplicity, the following analysis model is defined for the *PasswordType*:

- |Length < 5 AND OnlyNumbers := PINCode|
- |Length >= 5 AND Length <= 9 AND OnlyAlphabets := NormalPassword|
- |Length > 9 AND Length < 12 AND NumberOfDifferentSymbols >= 3 := GoodPassword|
- |ELSE := WeakPassword|

Thus, four base measures are used in this analysis model, i.e., *OnlyNumbers*, *OnlyAlphabets*, *Length* and *NumberOfDifferenSymbols*. These are measured using appropriate measurement methods, respectively (methods for *OnlyNumbers* and *OnlyAlphabets* are omitted from Fig. 4. In addition, these base measures are also connected to the AIS via appropriate attributes. For reasons of clarity, these are not presented in Fig. 4; however, TABLE II also lists these attributes.

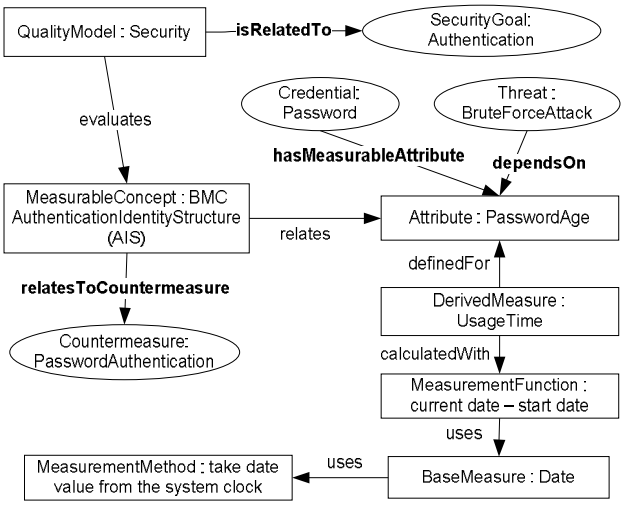

Figure 3. The age of the password.

The above mentioned attributes *PasswordStructure* and *PasswordAge* are mapped to the *BruteForceAttack* threat from the OIS. Thus, these are possible extension points in the future, in so far as risk related measures are added to ISMO. The risk measures are applied for run-time usage in [21].

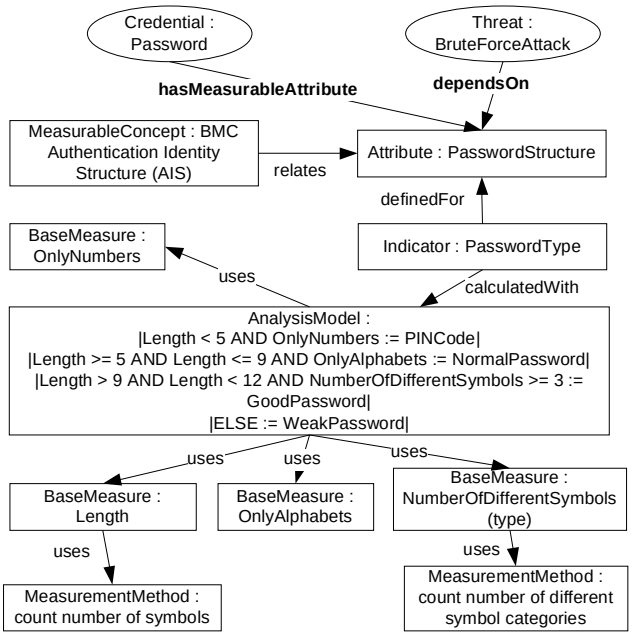

Figure 4. The structure of the password.

The final measure instantiated into the ontology is *AuthenticationLevel* (Fig. 5), which is an instance of the indicator concept – intended to combine the above described measures. The *AuthenticationLevel* is calculated with an analysis model in a similar manner as described for the *PasswordType* above. The analysis model is as follows:

- |PasswordType == PINCode := Level1|
- |PasswordType == NormalPassword AND UsageTime >= 180 AND UsageTime < 365 := Level2|
- |(PasswordType == GoodPassword AND UsageTime < 180) OR (PasswordType == NormalPassword AND UsageTime < 90 := Level3)|
- |ELSE := Level1|

The analysis model for the *AuthenticationLevel* uses the results from the *PasswordType* indicator and the *UsageTime* derived measure. Therefore, the calculation of the authentication level, according to this analysis mode, requires the five base measures, i.e., *OnlyNumbers*, *OnlyAlphabets*, *Date*, *Length*, and *NumberOfDifferent-Symbols*, presented in Fig. 3 and Fig. 4.

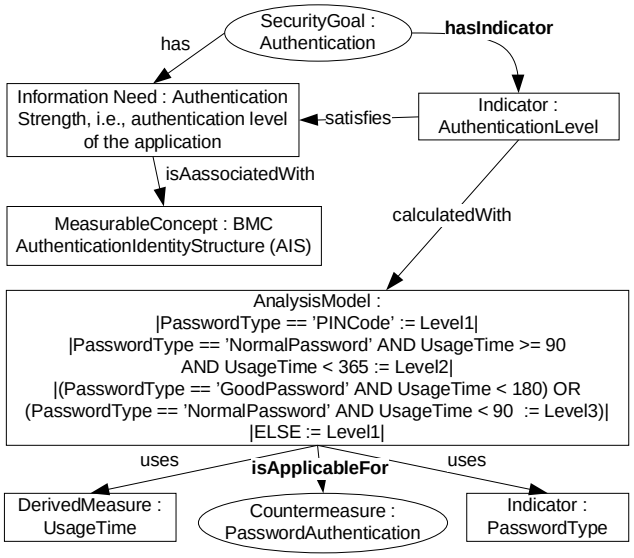

Figure 5. Authentication level.

Currently, the presented analysis models are very simple, utilising only a few base measures, and need to be enhanced in the future. Nevertheless, these analysis models provide the possibility to test the suitability of ISMO for run-time security measurements. Furthermore, the presentation of analysis models in the ontology makes it possible to modify and update them at run-time. The following table lists the mapping properties made from/to instantiated security measures. Again, these mappings are measure dependent. Hence, the addition of a new measure instance also creates new mapping properties.

TABLE II. MAPPING THE PROPERTIES OF INSTANTIATED MEASURES

Concept from OIS [7]	**Mapping property in ISMO (direction)**	**Concept from the SMO** [15]
Threat : Brute-ForceAttack	dependsOn (->)	Attribute : PasswordStructure Attribute : PasswordAge
Credential : Password	hasMeasurable-Attribute (->)	Attribute : PasswordStructure Attribute : PasswordAge Attribute : PasswordLength Attribute : NumberOfDifferentSymbols Attribute : OnlyAlphabets Attribute : OnlyNumbers
Countermeasure : Password-Authentication	relatesToCount ermeasure (<-)	Measurable concept : AuthenticationIdentity Structure
Countermeasure : Password-Authentication	isApplicableFor (<-)	Analysis model : analysis model for authentication level

V. THE USAGE OF INFORMATION SECURITY MEASURING ONTOLOGY

This section describes how the ISMO will be used at design and run-time. In addition, ontology evolution is discussed.

A. Utilisation at Design-time

The software designers have to take several issues into account when they design an application that is intended to measure its security level and adapt itself accordingly. Firstly, the required security goals are defined – such as the user authentication. Secondly, the levels for each security goal are defined, e.g., level 1 for security goals which are not very critical and level 5 for extremely critical security goals. It is notable that ISMO does not restrict the number of security levels, for example, the analysis models in the previous section utilised three levels instead of five. Thirdly, the security mechanisms to achieve the required goals are selected, e.g., a username-password pair for authentication. The micro-architecture for run-time security adaptation is presented in [3] – working as a guideline for the software developer by showing the components which are required in an adaptation applicable software.

The OIS already contains mapping properties from goals to supporting mechanisms. However, there is no possibility to define the required levels for the goals. It should be noted that the selection and implementation of security countermeasures is highly context-dependent. The ISMO draws a mapping from the security goal to the level indicator – in our case authentication level – as presented in Fig. 5. Therefore, the software designer can retrieve the base measures from ISMO, used for calculating a particular level indicator. Based on this information, she implements the measuring methods of base measures as the part of application. For example, the authentication level indicator requires five base measures and the related measurement methods as mentioned earlier, which must all be implemented to the application. However, measurement functions and analysis models that combine base measures are not hard coded to the application. Instead, a generic

 ISBN: 978-1-61208-130-4

parser and monitor components are utilised. The parser component retrieves analysis models from ISMO and parses the rules, which depict how the base measures have to be combined. The monitor component utilises these rules and calculates the security level (the authentication level at this time) from the base measures, which is utilised in the software adaptation. The generic and implementation specific parts are presented in Fig. 6. The internal design of these components is not within the scope of this paper.

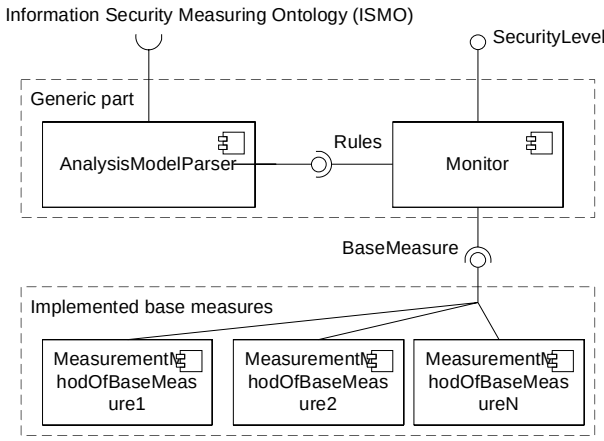

Figure 6. Generic and implementation specific parts.

B. *Utilisation at Run-time*

The application, which contains a capability to measure its security, is assumed to be aware of its security goal(s) and level(s) and how to measure the fulfilment of its goal(s). Hence, the application retrieves an indicator which is used to measure the level of a particular security goal, e.g., the authentication level indicator for the authentication goal. However, separate analysis models are required for the alternative countermeasures used to achieve user authentication. For example, Fig. 5 contains the analysis model for the password based authentication. Simultaneously, the ontology may contain an analysis model for the security token based authentication, and both of these analysis models are related to the authentication level indicator. Thus, the application must check the currently used countermeasure and select an appropriate analysis model for it from the ISMO. The *isApplicableFor* property maps the analysis model to the countermeasure and makes this selection possible.

Now, the application has a right analysis model. Based on this information, the application searches the measures that are used in the analysis model. The authentication level indicator and the related analysis model, presented in Fig. 5, use the *PasswordType* indicator and the *UsageTime* derived measure. Thus, the application queries ISMO until it finds the base measures which are required to calculate a value for the authentication level indicator. It is notable that these searches only have to be made at application start-ups and when the countermeasure is to be changed at run-time, due to security adaptation demands.

As a result of this search, the application possesses all the information which is required for measuring security. The application has the knowledge of required security goals and levels, and the base measures to be used. Therefore, the application uses measurement methods, which are implemented as a part of it during the design-time. The monitor component (in Fig. 6) combines these base measures to a security level indicator. In a situation where the application is unable to reach the required security level, it adapts the used countermeasure. The results of measuring help to recognise the part of the application that has to be adapted. The security adaptation is discussed in more detail in [21].

It is possible that the required security level changes during the application execution. For instance, the usage of the application may change in a way that requires a higher security level. This type of change does not affect the utilisation of ISMO or the measuring itself. Only the level, compared to the measurement result, changes.

C. *Ontology Evolution*

At some point, it is necessary to make changes and additions to ISMO. This is required because new threats appear, the usage of the application changes, or the environment of the application changes. The ontology evolution is a challenge from the application point of view, since ISMO is also used for making design decisions. In other words, the required base measures are selected and implemented at the design-time. Thus, a new base measure cannot appear for the application by adding it to ISMO. On the contrary, the analysis models which are used for indicators, such as the authentication level, can be dynamically changed to ISMO. For instance, the analysis model in Fig. 5 defines that level 2 is achieved with a normal password that is used for 3-12 months (90-365 days). However, this can be easily changed to the form: level 2 is achieved with a normal password that is used for 3-6 months. More complicated changes can also be made easily – the only requisite is that the application contains the required base measures. The *AnalysisModelParser* and *Monitor* components (Fig. 6) ensure that changes in the analysis models do not require any changes to the application. However, the analysis models have to be described in a common syntax that the *AnalysisModelParser* is able to parse. ISMO uses simple logical operations to combine the named measurement results, as seen in Fig. 4 and Fig. 5.

VI. A CASE EXAMPLE OF INFORMATION SECURITY MEASURING ONTOLOGY

Run-time security measuring and adaptation was earlier validated in [21, 22] – released on YouTube [23]. Now, a case example is used to exemplify both the design-time and run-time usage of ISMO. The case study takes place in a smart home environment, where the user performs different tasks with her mobile device. The RIBS platform [24] is used to build up the smart home environment. RIBS is a platform that makes it possible for heterogeneous devices to communicate with each other by a means of SIB (Semantic Information Broker) and agents. The SIB is an information

 ISBN: 978-1-61208-130-4

storage where agents publish and subscribe information. The smart home environment contains agents which publish environmental information, for instance, temperatures, humidity, etc. Home automation devices contain agents which subscribe to control information from the home SIB. Furthermore, the smart home environment contains agents, which offer entertainment information for the user, i.e., news, weather forecasts, etc.

In the case study, information from the smart home is utilised with a smart space application, which is running on a Nokia N900 mobile device. The smart space application and the related base measures are implemented using the Python programming language. In this case example, two SIBs exist. The first one (personal SIB) runs on the user's N900, as storage for ISMO. Alternatively, ISMO can be stored in the mass storage of the N900 in an OWL format. The second one (home SIB) runs on a computer in the smart home, and constitutes the home smart space. The application communicates with the personal and home SIB via TCP/IP communication and measures the achieved authentication level by a means of ISMO.

Firstly, ISMO is used at the smart space application design time as described in the previous section. The generic part, i.e., *AnalysisModelParser* and *Monitor* components, are imported to the application. The application developer makes a decision that passwords will be used for authentication and searches the supporting analysis model from ISMO. Furthermore, the base measures which are required in the analysis models are retrieved and implemented to the application. In this case, the used base measures are *OnlyNumbers*, *OnlyAlphabets*, *Length*, *NumberOf-DifferentSymbols*, and *Date*.

Secondly, ISMO is used while running a smart space application. When the user opens the smart space application, the application automatically retrieves ISMO from the personal SIB. The user then opts to join the home smart space with the smart space application. During the join operation, the user is authenticated for the first time and the authentication level monitoring process starts. The *AnalysisModelParser* component reads ISMO. The *Monitor* component receives rules on how to combine different base measures and provides the authentication level for the security adaptation. Both the *AnalysisModelParser* and *Monitor* components are running on the N900. The used countermeasure is password authentication – based on this information, the monitor component selects the correct analysis model to calculate the authentication level. It is notable that the application can contain several authentication mechanisms and ISMO provides information concerning which analysis model to use with each mechanism. The home smart space contains various types of information and the utilisation of different information requires their own authentication levels. Thus, we defined the following authentication requirements for different tasks:

- Level 1 for entertainment usage,
- Level 2 for retrieving information from sensors, etc.,
- Level 3 for controlling building automation devices.

The smart space application is aware of what the user is currently doing, i.e., it monitors the current context and reports this information to the security adaptation. The user decides to login with a username and password on authentication level 2. Thus, the user is unable to control the building automation devices. In an accelerated use case, when the password usage time reaches 12 months, the authentication level decreases to level 1. Hence, the smart space application only provides entertainment information for the user. When the user attempts to perform a task which requires higher authentication level, the smart space application recognises that an adaptation is required. The adaptation asks the user to re-authenticate with a better password, as is shown in Fig. 7. Consequently, the application user does not require any knowledge of ISMO, i.e., the smart space application seamlessly utilises the content of ISMO.

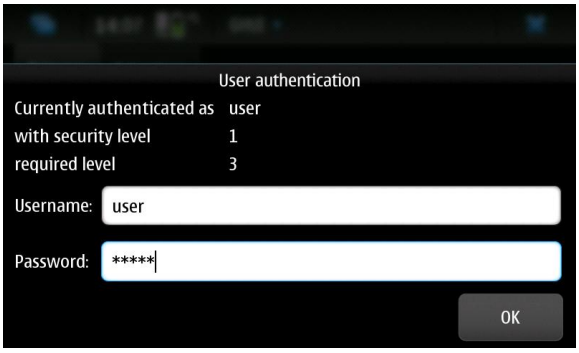

Figure 7. Re-authenticating the user.

The purpose of the case example was to test designed and instantiated ontology. Thus, the content of the analysis models and measures was not within the scope of the case. The utilisation of ISMO ensured that security can be measured in a dynamic environment. Without ISMO, the used analysis models have to be hard coded to the application, which is unreasonable in the dynamic environment. The case example proved that when the *AnalysisModelParser* and *Monitor* components exist, the implementation of the security measures to the application is straightforward. The application developer merely has to implement the required base measures as declared in the ISMO, or use existing base measures. The application developer is able to utilise measures from ISMO, without a need to implement ontology parsers. Moreover, the application was able to retrieve analysis models from ISMO and monitor the authentication level at run-time. The Monitor component calculates a new authentication level each time the used base measures change. However, the *AnalysisModelParser* component checks the content of ISMO at pre-defined intervals.

It is a commonly known issue that ontology searches may cause performance overheads. However, in this case example, the ontology was used in a mobile device without a major overhead. Nevertheless, it is important to optimise how often information is retrieved from ISMO. This helps to achieve the performance requirements of the application,

 ISBN: 978-1-61208-130-4

since changes in ISMO are only checked at pre-defined intervals. Therefore, there is no need to continually query the personal SIB. In the case example, the searches were made every 60 seconds and this kind of checking interval had no visible effects on the usability or performance of the application. Another alternative is to utilise subscriptions, which automatically inform to applications when ISMO is changed. However, the performance overhead of this option is not known beforehand, i.e., changes in ISMO can take place at anytime.

VII. DISCUSSION

In this work, we utilised existing ontologies – instead of starting from scratch – to achieve the information security measuring ontology for run-time usage. Thus, we gained a wide and extensible ontology that is compatible with its predecessors. The combination also ensures a higher maturity level, as the ontologies which were used were already validated. It can be seen from the ontology comparison presented in [10] that the existing security ontologies contain a large deal of overlapping. This work does not add overlapping concepts, which is important from a compatibility viewpoint. Utilisation of the SMO ensures that the measurement part of ISMO is generic. Therefore, the addition of new measures in the future will be easy. Furthermore, the used concepts can also be utilised to measure other quality attributes. Initially, the SMO is not intended for run-time usage. However, there are no constraints to applying the SMO at run-time situations as measuring related terminology is similar in both design and run-time measurements.

It might seem that using ontology to achieve a run-time measuring applicability is a too heavy weight solution. Nevertheless, in cases where an application contains several mechanisms for reaching a particular security goal, it will be necessary to describe the measures in detail. This is particularly necessary when the application is intended to adapt used security mechanisms. In addition, ISMO makes it possible to update and add analysis models – when a new vulnerability is found or the application usage changes. Currently, measurement functions and analysis models are described by using simple logical operations in the ontology – parsed by the *AnalysisModelParser* component. Logical operations were suitable for the measures used in this work. However, in the future, there is a need for additional mathematical operations, required in security measuring. The ontology definition is made at a level that possesses sufficient detail, and thus, it is possible for ISMO to provide the required knowledge for an autonomous measuring process.

Mapping between OIS and SMO is a complex task due to the complexity of measuring security. Currently, a concept level mapping is done, but there were only a few concept-to-concept mappings, which enforces the creation of mapping from/to instantiated security measures. Authentication related measures are instantiated to ISMO as an example. Additional mappings are required when a new measure instance is added. However, the measure instances added in this work offer an example of how to add the mappings, and thus, facilitate future additions. It is notable, that different types of measures will create entirely different mappings between these ontologies. For example, risk measures will create mappings between assets from the OIS and attributes from the SMO. On the other hand, there is not always a mapping property from the attribute concept (in SMO) to some specific credential (in OIS). Hence, mappings between these ontologies depend on the security goal, the used security mechanism, and the used measure.

The performed case example showed that ISMO can be used even in a mobile device without a major performance overhead. However, a more thorough performance evaluation has to be performed in the future. One question is how the usage of ISMO affects the achieved security. For instance, an attacker may cause constant environment changes, which in turn create a set of queries for ISMO and might jeopardize the availability. In addition, measurement methods and results might also be the target of an attack. Thus, it is necessary to perform the run-time measurement in a way that supports the achievement of security requirements, instead of creating new vulnerabilities and possibilities for attacks.

Survey of adaptive application security in [25] lists few adaptation approaches. Added to these, the Extensible Security Adaptation Framework (ESAF) [26] utilises security policies to adapt security mechanisms in the middleware layer. Furthermore, adaptation for Secure Socket Layer (SSL) is presented in [27][28] and monitoring for Java ME platform in [29]. The ISMO offers several advantages compared to existing self-adaptation and policy based approaches. Firstly, security measuring triggers an adaptation task, instead of beforehand defined situation. Secondly, ISMO is a generic solution, i.e., it is not tied to only one security mechanism, platform, or security goal. Finally, ISMO based approaches are dynamic – new analysis models can be added and the existing ones can be modified.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we presented a novel ontology – called ISMO – for information security measuring, developed particularly for the needs of run-time security measuring. The main purpose was to achieve an ontology that is able to support security measuring at the run-time of an application. The ontology development utilises two existing ontologies: (i) an ontology of information security, describing security related concepts, and (ii) a software measuring ontology, describing general measuring terminology. Firstly, a conceptual mapping between these ontologies was introduced. However, security measuring is a complex task where only a few concept-to-concept relationships can be made. Secondly, the ontology was instantiated by using password related measures. The measures which were used were simple – password structure and password age – however, these measures offered a good starting point to construct ontology which is applicable to run-time security measurements. After the ontology instantiation, we described how to utilise the ISMO in a way that supports run-time measurements. The case example was utilised to exemplify how to use ISMO in a smart home environment. Finally, we

 ISBN: 978-1-61208-130-4

discussed the advantages and shortcomings related to the designed ontology.

In the future, it is important to evaluate the performance cost of using ISMO. In addition, it is important to add new security measures to ISMO, and test how easily these extensions can be made.

ACKNOWLEDGMENT

This work has been carried out in the SOFIA ARTEMIS and GEMOM EU FP7 projects, funded by Tekes, VTT, and the European Commission.

REFERENCES

[1] D. M. Chess, C. C. Palmer, and S. R. White, "Security in an autonomic computing environment," *IBM Systems Journal, 42(1),* pp. 107-118. 2003.

[2] E. Ovaska, A. Evesti, K. Henttonen, M. Palviainen, and P. Aho, "Knowledge based quality-driven architecture design and evaluation," *Information and Software Technology, 52(6),* pp. 577-601. 2010.

[3] A. Evesti and S. Pantsar-Syväniemi, "Towards micro architecture for security adaptation," *1st International Workshop on Measurability of Security in Software Architectures (MeSSa 2010),* pp. 181-188, 2010.

[4] ISO/IEC 9126-1:2001. *Software Engineering - Product Quality - Part 1: Quality Model.* 2001.

[5] A. Avižienis, J. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing,* vol. 1, pp. 11-33, 2004.

[6] ISO/IEC 15408-1:2009, *Common Criteria for Information Technology Security Evaluation - Part 1: Introduction and General Model.* International Organization of Standardization, 2009.

[7] A. Herzog, N. Shahmehri, and C. Duma. (2009, "An ontology of information security," In *Techniques and Applications for Advanced Information Privacy and Security: Emerging Organizational, Ethical, and Human Issues,* Eds. H. R. Nemati, pp. 278-301, 2009.

[8] J. Zhou, "Knowledge Dichotomy and Semantic Knowledge Management," *Industrial Applications of Semantic Web,* pp. 305-316, 2005.

[9] C. Blanco, J. Lasheras, R. Valencia-García, E. Fernández-Medina, A. Toval, and M. Piattini, "A systematic review and comparison of security ontologies," *3rd International Conference on Availability, Security, and Reliability (ARES 2008),* pp. 813-820, 2008.

[10] A. Evesti, E. Ovaska, and R. Savola, "From security modelling to run-time security monitoring," *European Workshop on Security in Model Driven Architecture (SECMDA),* pp. 33-41, 2009.

[11] G. Denker, L. Kagal, and T. Finin, "Security in the Semantic Web using OWL," *Information Security Technical Report, 10(1),* pp. 51-58. 2005.

[12] A. Kim, J. Luo, and M. Kang, "Security Ontology for annotating resources," *LNCS,* vol. 3761, pp. 1483-1499, 2005.

[13] P. Savolainen, E. Niemelä, and R. Savola, "A taxonomy of information security for service centric systems," *33rd EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA 2007),* pp. 5-12, 2007.

[14] B. Tsoumas and D. Gritzalis. "Towards an Ontology-based Security Management," *20th Advanced Information Networking and Applications 2006 (AINA 2006),* pp. 985-992, 2006.

[15] F. García, M. F. Bertoa, C. Calero, A. Vallecillo, F. Ruíz, M. Piattini, and M. Genero, "Towards a consistent terminology for software measurement," *Information and Software Technology, 48(8),* pp. 631-644, 2006.

[16] N. F. Noy and D. L. McGuinness. "Ontology development 101: A guide to creating your first ontology,", pp. 1-25 2001.

[17] R. Savola, "A Security Metrics Taxonomization Model for Software-Intensive Systems," *Journal of Information Processing Systems, 5(4),* pp. 197-206, 2009.

[18] R. Savola and H. Abie. "Development of measurable security for a distributed messaging system," *International Journal on Advances in Security, 2(4),* pp. 358-380, 2010.

[19] C. Wang and W. A. Wulf, "Towards a Framework for Security Measurement," *Proceedings of the Twentieth National Information Systems Security Conference,* pp. 522-533, 1997.

[20] SPARQL Query Language for RDF, W3C Recommendation, http://www.w3.org/TR/rdf-sparql-query/, 31.1.2011

[21] A. Evesti and E. Ovaska, "Ontology-Based Security Adaptation at Run-Time," *4th IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO),* pp. 204-212, 2010.

[22] A. Evesti, M. Eteläperä, J. Kiljander, J. Kuusijärvi, A. Purhonen, and S. Stenudd, "Semantic Information Interoperability in Smart Spaces," *8th International Conference on Mobile and Ubiquitous Multimedia (MUM'09),* pp. 158-159, 2009.

[23] Semantic Information Interoperability in Smart Spaces, http://www.youtube.com/watch?v=EU9alk9t7dA, 31.1.2011

[24] J. Suomalainen, P. Hyttinen, and P. Tarvainen, "Secure information sharing between heterogeneous embedded devices," *1st International Workshop on Measurability of Security in Software Architectures (MeSSa 2010),* pp. 205-212, 2010.

[25] A. Elkhodary and J. Whittle, "A Survey of Approaches to Adaptive Application Security," *International Workshop on Software Engineering for Adaptive and Self-Managing Systems, 2007 SEAMS '07.*, p. 16, 2007.

[26] A. Klenk, H. Niedermayer, M. Masekowsky, and G. Carle, "An architecture for autonomic security adaptation," *Ann Telecommun, 61(9-10),* pp. 1066-1082. 2006.

[27] C. J. Lamprecht and A. P. A. van Moorsel, "Adaptive SSL: Design, Implementation and Overhead Analysis," *First International Conference on Self-Adaptive and Self-Organizing Systems, 2007. SASO '07.*, pp. 289-294, 2007.

[28] C. J. Lamprecht and A. P. A. van Moorsel, "Runtime Security Adaptation Using Adaptive SSL," *Dependable Computing, 2008. PRDC '08. 14th IEEE Pacific Rim International Symposium,* pp. 305-312, 2008.

[29] G. Costa, F. Martinelli, P. Mori, C. Schaefer, and T. Walter, "Runtime monitoring for next generation Java ME platform," *Comput. Secur., 29(1),* pp. 74-87. 2010.

 ISBN: 978-1-61208-130-4