We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,300

Open access books available

172,000

International authors and editors

190M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Security Trade-off – Ontological Approach

Bialas Andrzej

*Institute of Innovative Technologies EMAG
Poland*

1. Introduction

The chapter presents the ontology creation process on the security trade-off methodologies example. The trade-off in the security investments is focused on finding an optimal solution with respect to the defined criteria. It is performed by the balancing of effects of security measures, which usually are opposing or conflicting. In such situation, finding an optimal solution is difficult and depends on the features of the applied methods and the application domain needs and constraints that vary from one domain to another.

The general motivation of the works is to support the researches on the security trade-off processes and methods using the advantages and new possibilities offered by the ontological approach. The security trade-off methods, developed separately in different application domains to meet their specific requirements, are diverse and there are many crosscutting issues in them. It is difficult to assess whether the given method can be used in the neighbour application domain because there is no common understanding of the method features and possibilities. Choosing the adequate method for the given application domain remains a challenge. Sampling and analyzing the knowledge dealing with these methods and their features is needed. In such situation, the ontological approach can be helpful, though two aspects of this issue exist. The ontological approach will be used to gain knowledge about the existing trade-off methods and tools, and some of them may be ontology-based as well. Generally, the ontology represents explicit formal specifications of the terms in the given domain and relations between them. The creation of ontologies allows to analyze, share and reuse knowledge in an explicit and mutually agreed manner.

The chapter deals with the preliminary works whose aim is to elaborate the Security Trade-offs Methods Ontology (STMO) and the related knowledge base. The STMO ontology concerns the basic trade-offs methods, especially risk-centred, best fitted for the selected application domains and used to support trade-off decisions in the security investments in these domains. The far-reaching aim of this work is helping to find the right method for the given application. This is the key issue requiring broad knowledge about the methods features and possibilities. The elaborated ontology has been designed for security experts from different application domains to choose the proper methods and tools, and indirectly to support their decision processes. More efficient trade-offs bring different benefits to stakeholders, e.g.: saving funds, improving security, better acceptance of measures, better management, improving the competitiveness and market position, etc.

The chapter discusses the first iteration of the STMO ontology incremental development process, aiming at the prototype creation. This prototype enables other, more extensive and long-term works, requiring deeper collaborative researches in the security trade-off methodologies, whose results can be concurrently incorporated into the developed target ontology and the related knowledge base. The presented there ontology prototype can be viewed as a general framework encompassing selected concepts and relationships and be able to sample knowledge concerning:

- needs and requirements of the application domains with respect to the trade-off processes support,
- identified trade-off methods, including risk-based ones, and their features.

Acquiring the knowledge will allow to define the basic criteria concerning the best fitted method for the given application.

Summing up, the first, main objective of this work is to provide a prototype of the ontology-based framework and tool:

- for researchers, to facilitate researches on the security trade-off methods and tools, aiming at their improvements and laying out their development directions, and to facilitate sampling, analyzing the related knowledge and making it available to the interested parties,
- for users and other interested parties, enabling them access to the knowledge about security trade-off methods in a concise way.

The second objective of this chapter is to present a complete example of the ontology creation process using the basic knowledge engineering principles and the proven, open for general use tool.

The Section 2 of this chapter presents an overview of the existing works concerning the ontological approach applied in the discussed security domain. To better understand the Security Trade-offs Methods Ontology elaboration process, the Section 3 gives an introduction explaining how ontologies are created. The STMO building process is discussed in the main section of the chapter, i.e. in the Section 4. The Section 5 sums up the achieved results and specifies the planned future researches.

2. An ontological approach within the security domain

Ontologies were elaborated recently in many disciplines where “a common understanding”, “a common taxonomy” or “reasoning” are important, such as: web-based applications, medicine, public administration, biology and information security. The chapter concerns the latter issue. To get a common picture and background of the presented and planned research and development focused on the specialized security trade-off ontology creation, a short overview of existing works should be done.

Security ontologies are varied and express different issues in this domain. The first group of works concerns generally risk management issues. The paper (Ekelhart et al., 2007b) deals with the ISO/IEC 27001 standard implementation while the work (Ekelhart et al., 2007c) – quantitative risk analysis. Both deal with the knowledge base application supporting the information security management systems implementation. The similar issue, i.e. general aspects of security management is presented by (Martimiano & Moreira, 2005a). The work (Tsoumas et al., 2005) concerns the selection of controls while the work (Martimiano & Moreira, 2005b) discusses incident management issues. The paper (Atkinson et al., 2006)

presents ontology-centred technology risk management architecture for the banking sector. This architecture is used for knowledge modelling and integrating different software components. This is another example of the ontological approach to the risk management applications.

The paper (Elahi & Yu, 2008) focuses on a goal-oriented approach for modeling and analyzing security trade-offs and incorporating them into the requirements analysis and architecture design. Goal orientation is also used to structure knowledge to assist designers in making security trade-offs. The authors aim to develop the conceptual modelling framework in the form of an ontology, which provides a required foundation for automated and formal reasoning.

The work (Vorobiev & Bekmamedova, 2007) discusses security and trust ontologies, expressing the risk analysis issues, security algorithm taxonomy, security functions, attacks and defence, and trust. In the work (Kim et al., 2005) the authors specify the extensive NRL (Naval Research Laboratory) security ontology, encompassing subontologies concerning the security of services, security agents, information objects, security algorithms, assurance and credentials. Additionally, they discuss ontology integration issues.

The work (Yavagal et al., 2005) presents an ontological approach to the modelling of the Common Criteria (ISO/IEC 15408) security functional requirements and their mapping to the specified security objectives. The work (Ekelhart et al., 2007a), related to the ontology available at the (Secure Business, 2008), is focused on the ontological representation of the Common Criteria (ISO/IEC 15408) security assurance requirements. The presented tool supports evaluators during the certification process in such activities like: planning an evaluation process, reviewing relevant documents or creating reports. The Author's papers concern also the Common Criteria methodology, presenting the Security Targets Ontology (Bialas, 2008) and the selected issues dealing with the security problem definition and solution using the introduced Specification Means Ontology (Bialas, 2009).

The examples of common security issues ontologies (CSL, 2008; DAML, 2008; Herzog, 2006; REI, 2008) can be analyzed using the Protégé Ontology Editor and Knowledge Acquisition System (Protégé, 2008). Please note the latter one (REI, 2008) presenting the REI ontology (precisely: the set of subontologies) which is used for the security policy development.

The review shows that the basic information security areas are represented by different ontologies. Their number is growing. Some of the mentioned ontologies concern risk management, even the security trade-off, but they are used as tools, e.g. supporting directly the risk management processes. None of these works is devoted to build the risk management methods taxonomy, trade-off methods taxonomy or the knowledge bases, classifying them, ordering their properties and helping to choose the right method for the given application.

Please note that there are some communities of professionals, e.g. (ENISA, 2009), (SARMA, 2009), which sample and manage the knowledge dealing with the risk management methods, but they do not use the ontological approach.

3. Ontology elaboration methodology and the used tool – introduction

In the computer science and information science, ontology is a formal representation of a set of concepts within a domain and the relationships between these concepts. It is used to

reason about the properties of that domain, and may be used to define the domain. (Wiki/ontology, 2009).

The ontology development, based on the basic knowledge engineering principles (Noy, 2001), begins with its domain and scope definition. It requires to investigate the matters that the ontology concerns (here: different aspects of the trade-off support). It allows to define the ontology competency questions (i.e. the questions that the ontology-related knowledge base is able to answer) and to identify the ontology terms within the domain. On this basis the definition of the hierarchy of classes and its properties can begin. The class hierarchy creates the taxonomy of terms in the discussed domain. Some classes have only an abstract meaning, some have instances, called also individuals. The ontology development needs permanent tests and finally the validation by ontology users. This way a prototype of the Security Trade-off Methods Ontology and the related knowledge base has been elaborated. As the development tool, the Protégé Ontology Editor and Knowledge Acquisition System from Stanford University is used (Protégé, 2008). The STMO will be expressed using the OWL (Web Ontology Language) language, precisely the OWL-DL (DL – Description Logics), which allows automatic reasoning, e.g. to compute the classification hierarchy.

4. Development of the Security Trade-off Methods Ontology (STMO)

The section presents the Security Trade-off Methods Ontology (STMO) elaboration process. Particular subsections discuss one step of this process, starting from the ontology domain and scope identification, through the concepts, properties and instances defining, to the ontology testing, validation and use. The STMO elaboration has two objectives:

- providing the common taxonomy of the security trade-off methods used in different application domains; allowing to better understand terms and relationships and to sample basic knowledge about the domain (the near objective, dealing with the STMO prototype presented in the chapter),
- creating on this basis the extensive knowledge database which can help to recommend the right method to solve the given problem in the considered application domain (the further objective, whose fulfilment is enabled by the STMO prototype).

4.1 Domain and scope of the ontology – determining the competency questions

The domain of the STMO ontology are the security trade-off methods, their features and applications, including related tools and their theoretical foundations. Please note that the trade-offs in security investments go beyond traditional cost-benefits and risk management approaches, though the risk issues still play the key role.

The analysis of the Security Trade-off Methods Ontology domain is focused on the following main issues:

- considered domains of applications,
- security trade-off methods,
- security measures implied by these methods,
- expected stakeholders' benefits of the use of the particular methods,

which are briefly discussed. The results of this discussion allow to identify important terms and to define the STMO concepts.

The targeted STMO ontology should consider a broad range of application domains where the security trade-off methods are used, e.g.:

- information security domain,
- financial sector (banking, insurance),
- critical infrastructures (transportation, telecommunications, fuel and energy),
- security-sensitive industry,
- public areas (citizens, governments, society),
- first responder organizations, anti-terrorism and anti-crime activities,
- health care sector,
- regulators (national; international),
- small business and others.

The concepts and relationships of the STMO ontology should express stakeholders' needs and requirements dealing with the security investments that can be encountered in these domains. The created ontology prototype will be open for any application domain but will be focused on the information security domain. Information technologies are the basis of business processes in other application domains, and their inherent risk has been a well known problem for years and for this domain many methods and tools have been developed. Other domains will also be considered though on a more general level of details. The concepts and relationships of the developed ontology should express different groups of methods supporting the trade-off processes, like:

- methods for risk management,
- methods for cost-benefit analysis,
- methods and frameworks for trade-off evaluation and decision support,
- other approaches, like scenario-based techniques, expert systems, process analysis, simulations & gaming, human factors models, etc.

The right method should provide the right (i.e. optimal with respect to the defined criteria) measures to solve the given problem within the considered application domain. Trade-offs has multidimensional character. For this reason, apart from the risk management issues including risk transfer and avoidance, the developed ontology should consider other factors, like: mitigation of uncertainties, impacts and cascading effects, trust, acceptance of measures by stakeholders, cross-cutting issues, integration into the regular systems or operations, supporting business processes, legal aspects, and different constraints. Security investment encompasses different kinds of the security measures used by the trade-off process. Sometimes they provide synergy but usually their impact is mutually opposing or even conflicting. Sometimes the measures implications have a financial or equivalent-to-financial dimension, sometimes they are difficult to express. The examples of the measures are:

- implementation of the selected technological solution enforcing security,
- fault-tolerant mechanisms, including redundancies,
- working out long-term security strategies based on the advanced risk management,
- regulations and legal acts,
- organizational and procedural measures,
- education, training and raising awareness.

The measures proposed by the methods used in the given application domain should bring different kinds of benefits for stakeholders, like:

- saving investment funds,
- gaining more financial return on security investment,

- improving their own security, competitiveness, market position, and supporting a long-term security strategy,
- selecting the best measures from alternatives and facilitating acceptance of the measures,
- prioritizing and scheduling decisions or selected measures,
- giving support to regulatory decisions and processes,
- improving confidence of societies in security measures, etc.

The scope of the STMO ontology is related to the ontology competency questions. They will be constructed around the main question: "How to find the right method to solve the given problem in the considered application domain?". The ontology-related knowledge base should be able to give an answer to this question (after defining the appropriate criteria) and to give an answer to the related, more precise questions, like: "Which is the method dealing with the anti-terrorism that considers escalation effects?", "Are there any goal-oriented methods considering spill-over effects?", "Which methods used for public areas of applications help to facilitate measures acceptance and provide detailed ROSI (Return on security investment) analysis results?", etc. Details of these answers depend on the maturity of the ontology and the gained knowledge. To solve this competency questions problem, a top-down method will be applied, beginning from the general issues, through step by step refining, to more precise issues.

4.2 Possible reuse of existing ontologies

During the ontology development the use of other, third-party ontologies should always be considered. The key issues are the range of compatibility, integration ability, quality, satisfied needs of the ontology users (who successfully validate or use it) and, first of all – the availability of the given ontology. The ontology reusability issue needs further investigation to better meet the needs and expectations of stakeholders.

On the discussed prototype stage of the STMO ontology development, no other ontologies will be reused. The open issue is to integrate, in the future, the STMO ontology with the external third-party ontologies, if they appear.

4.3 Identifying important terms in the ontology

The basic set of important terms encompassed by the STMO ontology is identified during the ontology domain analysis. The terms are grouped by domains of applications, used methods, proposed measures and their effects, as it was mentioned earlier in the subsection 4.1. They are supplemented by more detailed issues identified during the preliminary analysis of some representative methods. These terms are used to define the ontology concepts (classes) and their properties.

4.4 Ontology classes (concepts) and the class hierarchy

Generally, a top-down ontology development approach is applied, though the entire process was iterative and some bottom-up activities were undertaken. In order to assure the proper knowledge representation, it is necessary to perform the analyses of terms and relations between terms, e.g.: class-individual, class-subclass, class-superclass, abstract classes and classes having instances. The important decision to make is about what should be expressed by a class and what by a property. It is also necessary to take into

consideration the possibility of the future evolution of class hierarchy, transitivity of class relations, avoiding common errors, naming convention, etc. Most class names (and properties names – subsection 4.5) are self explanatory, though their lengths should be limited.

For the STMO ontology prototype all concepts are grouped in the following main classes:

- the AppDomain class, representing different application domains,
- the SecMeasure class, representing security measures implied by methods,
- the Benefit class, expressing stakeholders’ security benefits (positive effects) of the implemented measures,
- the MeasSideEffect class, expressing stakeholders’ side effects (negative effects) of the implemented measures,
- the STOMethod class, expressing investigated security trade-off methods,
- the AuxiliaryConcept class, representing a set of auxiliary and diversified terms, whose individuals are used for main classes defining and the knowledge organizing and retrieving.

The first of the main classes, the AppDomain class, represents basic application domains of the security trade-off methods (and their subdomains, when needed). This class has subclasses that all together define the taxonomy of the considered application domains (Table 1).

Class	Subclass	Meaning
ATerrCrimeAD		Anti-terrorism and anti-crime activities
CritlInfraAD		Critical infrastructures and their protection
	FuelEnerg	Fuel and energy sectors
	TeleComm	Telecommunications
	Transport	Transportation
FinancialAD		Financial sector
	Banking	Banking
	Insurance	Insurance
FirstRespAD		First responder organizations
HealthAD		Health care sector
InfoSecAD		ICT and information security
PubAreasAD		Public areas, including the citizens, governments, society
RegulatorAD		The regulators – organizations functioning on the national or international levels
SensIndustryAD		Security-sensitive industry
SmallBusinAD		Small business
OtherAD		Other, unclassified application domains

Table 1. Application domains (the AppDomain STMO ontology main class subclasses) – note that a subclass is also a class

This application domain taxonomy has a preliminary and open character. It will be refined and extended on the basis of the validation results and gained experience during the STMO ontology use. Probably, more subdomains for the public areas, financial sector, first responders, critical infrastructures ought to be distinguished.

The second main ontology class, i.e. the **SecMeasure** class, represents different kinds of the security measures implied by the considered methods, shown in the Table 2.

Class	Meaning
AdvancedTech	The implementation of the advanced technologies or infrastructure solutions
CoopFramew	The organized cooperation frameworks, such as: national or transnational cooperation, public-private partnership, emergency teams, exchanging the information about vulnerabilities, threats, risks, new technology solutions, sampling statistics, etc.
DevelopStrategy	Activities based on the elaborated security strategies, especially the long-term strategies
EduAware	Education, trainings and raising awareness activities
OrgProced	Different forms of the organizational and procedural measures
Redundancy	Different types of redundancies built-in to the systems
RegLegisStd	The regulation, legislation and standardization activities
RiskMngmt	The risks management implementation

Table 2. Security measures representation (the **SecMeasure** subclasses)

The third main ontology class, i.e. the **Benefit** class, expressing stakeholders’ benefits, as the results of measures implied by methods, includes the issues shown in the Table 3.

Class	Meaning
AvoidEscalation	Disabling different forms of the threat escalation effects
FacilMeasAccept	Facilitating the selected measures acceptance
ImprovBussPosit	The general improvement of the business position, e.g. the competitiveness or market position
ImprovOwnSec	The improvement of the security of the stakeholder’s organization, e.g. decreasing the number of incidents and related damages
ImprovPrivacy	The improved confidence of societies in security measures
LongTermSupp	Supporting the long-term business strategy
MonetaryReturn	More financial return on security investment
PosSocietImpact	The positive societal impacts of the security trade-off decision
PrioritizSchedul	The possibility to schedule and prioritize the implied security activities
RegDecisInfluence	The possible influence on the regulatory decisions and supporting the regulatory processes
RiskMngmtEffect	The risks management effects, e.g.: risk mitigation, risk transfer to a partner or insurer
SaveInvesFund	The saved investment funds
SelectOptions	The measures selection from the set of possible options
SpillOver	Additional, positive and indirect effects of the investment expenditures

Table 3. Possible benefits considered by methods – an ontological representation (the **Benefit** subclasses)

Apart from the benefits implied by the security measures, also their negative impacts (side effects) should be considered. They are expressed by the **MeasSideEffect** subclasses, shown in the Table 4.

Class	Meaning
DecrOperAbility	The decreased operational ability (the business efficiency lowers)
DecrPrivacy	The decreased privacy
DecrSecurity	The decreased security (e.g. in other areas)
ExceedInvesFund	The exceeded investment funds
LessReturn	The lower than expected financial return on security investment
NegBusinImpact	The negative impact on the business processes, increased business costs, decreased products quality, new barriers, obstacles as the consequences of the applied measures
NegSocietImpact	The negative societal impacts of the security trade-off decision

Table 4. Possible side effects considered by methods – an ontological representation (the **MeasSideEffect** subclasses)

The most important, fifth main class, i.e. the **STOMethod**, represents the investigated security trade-off methods. This is the central point of the entire STMO ontology. Currently, there are no **STOMethod** subclasses defined. Particular kinds of methods are distinguished by the **STOMethod** property, represented by the **MethCategory** class that will be explained later. The results of first investigation of the methods demonstrate that it is difficult to classify the given method into one category. Most of them have a complex, mixed-mode character. In the future, after investigating the representative number of methods and defining the criteria allowing to recommend given methods for particular application domains, some subclasses of the **STOMethod**, like: **Best4ATerrCrimeAD**, **Best4CritlInfraAD**, etc. can be defined as the inherited classes using the Protégé built-in reasoning facilities.

The sixth main class, the **AuxiliaryConcept** class, includes varied issues. The individuals of the **DecMakingLev** class express different decision making levels (tactical, operational, strategic, national level, etc.). The **MeasureScale**, having subclasses: **MonetaryScale**, **QualitScale**, **QuantitScale**, is used to define different kinds of measurement or assessment scales used by the security trade-off methods.

A very important class is **MethCategory** which presents the assumed basic taxonomy of the investigated methods, containing the following subclasses:

- the **CostBenMeth** class – encompasses the cost-benefit approach methods, including:
 - the **CostBenAnalys** class – concerns the cost-benefit analysis,
 - the **CostEffectAnalys** class – is related to the cost-effectiveness analysis,
 - the **MultiFacAnalys** class – deals with the multi-factor analysis,
- the **DecSuppMeth** class – is related to the trade-off evaluation and decision-support, including:
 - the **AspectOriented** class – concerns the aspect-oriented and risk-driven methods,
 - the **CautPrecautPrinc** class – deals with the methods based on the cautionary and precautionary principles,

- the **DecisAnalSupp** class – deals with the decision analysis and decision support methods,
 - the **DesignOriented** class – concerns the design-oriented methods,
 - the **GoalOriented** class – deals with the goal-oriented methods,
 - the **MissionOriented** class – deals with the mission-oriented methods developed in the military sector,
- the **RiskAssMeth** class – deals with the risk analysis and assessment methods, i.e.:
 - the **CmplexRiskAnalys** class – is related to the complex risk analysis methods,
 - the **ROIbasedAnalys** class – is related to the ROI-based (Return on investment) methods,
 - the **ROSIbasedAnalys** class – deals with the ROSI-based (Return on security investment) methods,
- the **OtherMeth** class encompasses other approaches, i.e.:
 - the **ExpertSysBased** class – is related to the expert systems use,
 - the **HumFacBased** class – deals with the methods based on the human factors models,
 - the **InterdepNets** class – is related to the interdependency networks for critical trade-offs including hard and soft factors,
 - the **OntologyBased** class – is related to the ontology-based methods,
 - the **ProcessAnalys** class – deals with the process analysis methods,
 - the **ScenarioBased** class – deals with the scenario-based techniques,
 - the **SimulGame** class – is related to the simulations & gaming,
 - the **Technical** class – is related to the technical domain (safety) methods adapted to the security applications.

It was assumed that a given method can belong to a few method categories.

The **RiskCategory** class expresses different kinds of risk considered by the investigated methods. The categories are represented by the subclasses having self-explanatory names: **BlackMailing**, **BusinProcDisrupt**, **CustomConfidLoss**, **Epidemy**, **Harming**, **HostageTaking**, **Injuring**, **KeyPersLoss**, **LiveLoss**, **MarkShareLoss**, **MngmtProcDisrupt**, **NaturDisaster**, **Riots**, **Spionage**, **Strikes**, **VitalSuppliesLoss**.

The **MethDevelOwner** class represents the person or the organization who has developed a method or owns it.

The **OrganizType** class concerns organization types (commercial, public, non-profit, governmental agencies, transnational organizations, etc.) taken into consideration by the methods.

The **SecInvArea** class deals with the considered investment areas, expresses by the subclasses: **Equipment**, **InfrastrucProt**, **OperatManag** (i.e. operational management), **Personnel**, **Services**, **StrategyPlan** (i.e. development of strategies and plans).

The **SecInvType** class concerns security investment types (new investment, expansion, modernization, etc.) taken into consideration by the methods.

The **SuppTool** class points at the supporting tool related to the method. This class needs refinement because currently only the basic information concerning the tool is sampled (name, developer, description, status of development, applications and usability).

The **ThreatEscalEffect** class expresses different kinds of threats effects considered by the methods (secondary, cascading, escalating, etc).

The **TimeHoriz** class concerns different time horizons (of the investments and their positive and/or negative impacts) taken into consideration while the given method is used.

The key-importance class, **TradeOff**, represents different elementary security trade-offs considered by the method. Each of them considers two factors (sometimes groups of factors) from those represented by the **TradeOffFactor** subclasses: **BusinessAcceptance**, **Convenience**, **Cost**, **EnvironAspect**, **EthicalAspect**, **LegalStdRestrict**, **MethodologyAcceptance**, **PoliticalAspect**, **Privacy**, **Prosperity**, **Security**, **SocialAcceptance**, **SocialCohesion**, **TechnologyAcceptance**, **Transparency**. The security factor expressed by the above mentioned **Security** class is of the key significance for the security trade-off decisions. Please note that the **AuxiliaryConcept** class is used to arrange different issues concerning the key ontology class, which is the **STOMethod** class.

The above presented taxonomy has an initial meaning. It is assumed at the beginning of the iterative STMO ontology development process. It will be revised during the knowledge acquisition (investigating details of terms embraced by this ontology) and the validation results (checking if the facilities and features offered by the ontology are useful). The revision, permanent improvement, can be enabled after finishing the first step of the ontology creation iterative process, i.e. after defining a certain number of class properties, some individuals, some queries, etc.

It was assumed that the lowest level classes can have instances (individuals) representing particular application domains, security trade-off methods, applied measures and achieved effects.

The presented there ontology developer's defined class hierarchy is called "an asserted class hierarchy". On this basis, and with the use of the reasoning mechanisms, "an inherited class hierarchy" can be constructed, when needed. It will be used to define best fitted methods for particular application domains.

4.5 Ontology class properties and their restrictions

The classes embraced by the defined hierarchy of classes can have some properties assigned, called slots (or roles). The slots are subject to some general principles. All subclasses of a given class inherit the slots of that class, therefore a slot representing the given class property should be placed on the highest possible level of the class hierarchy. Besides, when a class has multiple superclasses it simply inherits slots from all of them.

There are three kinds of standard slots (Noy, 2001):

- object-type (also called "instance-type") slots; they are used to express "complex properties", i.e. relationships between an individual member of the given class (the object) and other individuals, e.g. when the given individual consists of other individuals or points to other individuals;
- data-type slots; they are used to express "simple properties" or "attributes", i.e. intrinsic or extrinsic properties of the individuals of the most elementary classes; the data type used for this slot can be any of the data types commonly used in the modelling or programming, e.g.: integer, byte, float, time, date, enumeration, string;
- annotation slots; they are used to express the meaning of the given concept, to document different ontology items (concepts, slots, individuals); annotation slots are RDF-based (RDF means Resource Description Framework); the most known

example of this slot is the `rdfs:comment` slot which gives more explanation of the given ontology item.

An instance-type slot is attached to the classes which are called a domain, while the classes indicated by this slot are called a range. The possible values of the slot can be refined by defining the restrictions (called also “facets”) for them. The restriction specifies or limits the set of possible values for the given slot.

The presented there STMO ontology uses all types of slots, however, the chapter is focused on high-importance slots describing key properties of classes. A typical situation deals with building the ontology for the insufficiently known domain. In this case the important issue is, when to use a data-type slot and when an object-type one for some properties. At the beginning the data-type slots can be used. After sampling and analyzing the knowledge, a better knowledge expression will be possible and some data-type slots can be replaced by object-type slots, which allow the better knowledge modelling and retrieving. Similarly to the classes, it was assumed that a slot name is related to its meaning.

With respect to the chapter objectives, the most important is the `STOMethod` class which represents the investigated methods. Some general-meaning, descriptive slots of the `STOMethod` have range strings. Other slots of the `STOMethod`, representing more structured knowledge, where pointing at some other individuals is possible (object-type slots), have a range of appropriate classes. The slot ranges are given in brackets, below. In the validation progress, some data-type slots can be replaced by object-type slots. The `STOMethod` class slots were organized by thematic groups.

The group of slots expressing the basic method features encompasses:

- the `rdfs:comment` slot is used as the reference to the source of information about the method,
- the `methAcron` slot (range: string) presents the used acronym (short name),
- the `STOMethodName` slot (range: string) presents the full method name,
- the `methDevelOwner` slot (range: `MethDevelOwner`) specifies the developer/owner of the method,
- the `STOMethodGenDescription` slot (range: string) contains a short method description,
- the `availabTool` slot (range: `SuppTool` class) specifies the tools supporting the given method, as the separate individuals with their own slots,
- the `appDomains` slot (range: `AppDomain` class) lists application domains in which the method can be used,
- the `seclnvesAreas` (range: `SeclnvArea` class) specifies the considered investment areas,
- the `organizType` slot (range: `OrganizType` class) presents the types of organizations who can use the given method,
- the `measurmScale` slot (range: `MeasureScale` class) lists the scales (qualitative, quantitative, monetary) used for the measurements or assessments of the expense values, gained benefits, encountered side effects – related to the trade-off,
- the `methCategs` slot (range: `MethCategory` class) specifies categories to which the given method can be classified,
- the `riskCategs` slot (range: `RiskCategory` class) lists categories of risk considered by the method,

- the **secTradeOffs** slot (range: **TradeOff** class) specifies security trade-off cases considered by the method; each case (the individual) expresses an elementary trade-off issue, e.g. **Security-Cost**, **Security-BussinAccept** (i.e.: security vs. business acceptance), **Security-Performance** (see details below),
- the **standardCompliant** slot (range: string) specifies main standards with which the method complies,
- the **methodOutput** slot (range: string) describes how the method results are presented for its user.

The group of slots expressing the investment features considered by the method and applied measures encompasses:

- the **expenditureDesc** slot (range: string) which generally describes the expense values considered by the method and related to the analyzed trade-off,
- the **secMeasures** slot (range: **SecMeasure** class) which lists the possible to use security measures,
- the **decisMakingLev** slot (range: **DecMakingLev** class), describing generally considered levels where the security trade-off decisions are undertaken, like: infrastructure management level, strategic level, project level, country level, etc.
- the **diffFrmOtherMeasures** slot (range: string) which expresses the differences between the proposed security measures and other measures, e.g. governmental, industrial (range: string),
- the **seclnvesTypes** (range: **SecInvType** class) which specifies investment types possible to consider, such as: new investment, expansion, modernization, etc.,
- the **investTimeHor** slot (range: **TimeHoriz** class) which lists the considered time horizons dealing with the investment, such as: 1 Year, 5 Years, etc.
- the **investSizeDesc** slot (range: string), presenting generally the investment sizes considered by the method,
- the **diffFromOtherInvestms** slot (range: string), expressing generally specific issues or the differences of the considered security investments from other kinds of investments e.g. safety-, defense-, and business-related investments,
- the **no-InvestmConseq** slot (range: string), which specifies the method ability to describe the consequences of no investment,
- the **organizatReqsDesc** slot (range: string), which presents organizational requirements for the effective use of the method.

The group of slots expressing the positive and negative effects of the investment considered by the method encompasses:

- the **consideredBenefitDesc** slot (range: string), generally describing the benefits considered by the method and related to the analyzed trade-off,
- the **midTermBenefs** slot (range: **Benefit** class), listing mid-term benefits possible to gain from the investment, considered by the method,
- the **longTermBenefs** slot (range: **Benefit** class), listing long-term benefits possible to gain from the investment, considered by the method,
- the **escalEffects** slot (range: **ThreatEscalEffect**) which specifies the threats escalation effects considered by the method, e.g. cascading effects,
- the **sideEffects** slot (range: **MeasSideEffect** class) which specifies the side effects related to achieved benefits, considered by the method and related to the analyzed trade-off.

The last group of slots is related to security trade-off method evaluation and includes:

- the `statusOfDevelopm` (range: string) – specifies the status of the method or tool development (newly created, prototype, matured, etc.)
- the `usabilityAssessm` slot (range: string) – which characterizes briefly complexity, friendliness, price, cost of analyses, required expertise, etc.,
- the `statusOfApplication` slot (range: string) – characterizes briefly the current status of applications and gained experiences with the use of the method or tool, e.g.: number of users, organizations, sectors, etc.,
- the `ifCombineQualitQuantAppr` slot (range: Boolean) – says if the method is able to combine qualitative and quantitative approaches while trade-off,
- the `ifMultipleFactors` slot (range: Boolean) – says if the method is able to measure or assess multiple effects and/or consider them while trade-off,
- the `ifDiffUserRoles` slot (range: Boolean) – says if the method supports different roles/groups of roles of users, varied working teams, etc.,
- the `strengthsOfMethod` slot (range: string) – generally describes the strength of the method considering its applicability conditions,
- the `weaknessOfMethod` slot (range: string) – generally describes the weakness of the method considering its applicability conditions,
- the `whatToImprove` (range: string) – generally describes what to improve in the method with respect to its applicability conditions.

Going back to the mentioned `TradeOff` class which expresses the security trade-off cases considered by method, please note the following slots of the `TradeOff` class:

- the `GenDescripOfSTOcase` slot (range: string) – generally describes the considered trade-off decisions, including conflicts, synergies and interdependencies between security and other issues,
- the `STOfactor1`, and the `STOfactor2` (both of the range: `TradeOffFactor`) – specify two groups of trade-off factors, generally mutually opposing, but the members of the given factor slot are mutually supporting; for the given trade-off problems it will be usually “1factor to 1factor”, but the combination “m factors to n factors” is also possible, especially to express more complex cases;
- the `STOpairsOfSynergyFactors` slot (range: string) – includes comments about the mutual supporting factors, describing their synergies,
- the `STOpairsOfOpposingFactors` slot (range: string) – includes comments about the opposing or even conflicting pairs of factors,
- the `STOinterdependencies` slot (range: string) – specifies interdependencies between elements of the considered groups of factors.

Please note that only the main slots were mentioned above, especially the slots related to the investigation of the trade-off method and trade-off concept. In the progress of the method investigation some slots will be refined or substituted by the better matching slots. It is an iterative process based on the sampled knowledge about different security trade-off approaches during the validation.

4.6 Creating individuals – instances of classes and filling in their slots

The next step of the ontology elaboration is to define the individuals, called also class instances. The instances represent class members. Please note that some classes have

instances, some not. General rules of the ontology engineering differ a little from the rules of the object modelling domain. Please note that an instance of a subclass is an instance of a superclass. For the Security Trade-off Methods Ontology the individuals are defined mainly for the classes of the lowest hierarchy level. The main set of individuals with filled-in slots, representing particular methods, is the core element of the STMO related knowledge base. The ontology is provided by different kinds of the auxiliary individuals necessary at the beginning of the security trade-off method investigation. They represent basic issues used to specify the investigated methods, including predefined values for some issues or patterns, showing how to define other individuals during this investigation. The STMO ontology creation process will be exemplified using the Protégé Ontology Editor and Knowledge Acquisition System (Protégé, 2008) on a few examples. The Fig. 1 presents one of them dealing with the creation of the TradeOff class individuals expressing trade-off cases. It was assumed that the given method is able to include many cases, placed in the secTradeOffs slot of the STOMethod class individuals. The Protégé tool is provided with the several tabs dealing with the metadata, OWL classes, properties, individuals, forms, queries and other functionalities. Let us consider the tab representing individuals.

Fig. 1. The STMO individuals related to the trade-off cases in the Protégé (Protégé, 2008)

The left side of the Fig. 1 ("Class Browser") encompassing a part of the STMO ontology class hierarchy and showing some AuxiliaryConcept subclasses related to the discussed trade-off cases. The numbers of the defined individuals of the given class are placed next to the class names within the brackets. For the highlighted TradeOff class their individuals (marked as rhombuses) are presented within the "Instance Browser" window in the middle: from Security-Availability to SolutionEffect-SolutionCost, representing particular trade-off cases predefined for different trade-off methods. In the right part of the figure "Individual Editor"

shows the highlighted `SolutionEffect-SolutionCost` case details, expressed by its slots. Please note `rdfs:comment` and other slots of the `TradeOff` class, explained at the end of the subsection 4.5. Please note the `TradeOffFactor` subclasses (on the left side). The combinations of their individuals are used to predefine needed trade-off cases, usually between security and other factors.

The most important individuals that constitute the STMO ontology-related knowledge base are `STOMethod` class individuals, characterizing particular methods. The ontological representation of the selected methods will be shown in the next subsection.

4.7 Testing and validation of the developed ontology

Similarly to other ontologies, the created STMO ontology should be tested on the fly and validated by the people who can use it. The Protégé environment provides some facilities to support both these activities.

While building an ontology, certain commonly known errors (Noy, 2001) may occur, such as: cycles in the class hierarchy, violation of property constraints, incorrectly defined terms, classes with a single subclass, classes and slots with no definitions, slots with no constraints like value type or cardinality, interval restrictions issuing empty intervals, e.g. when `min val > max val` was assigned. They can be detected with the use of the Protégé menu functions, like “Checking consistency”, “Run ontology tests”, or by manual ontology inspections supported by the Protégé built-in visualization tool (OWLViz). While elaborating the ontology, some usability tests can be performed on the fly to early detect some of the above mentioned errors. The ontology developer checks if the right structures of individuals are composed, if they have assumed properties, if the needed information can be retrieved properly by queries from the knowledge database, and if the forms were properly defined.

The ontology development process has a subjective character because many different correct solutions are possible, as well as an objective character because the applied solutions should be objectively correct (Noy, 2001). Please note that usually not all of the possible correct solutions met the expectations of the ontology users. For this reason, the ontology validation is important, which helps to reshape the ontology according the users’ requirements. The ontology project ought to be flexible and open to changes. During the ontology validation process the class hierarchy, slots and defined forms are considerably rebuilt on the fly.

The STMO ontology validation encompasses the basic scenarios of the ontology use. Currently, the main scenarios deal with the knowledge acquisition and its retrieving for the ontology testing purpose:

- creating the individuals describing identified properties of the reviewed security trade-off methods examples,
- sampled knowledge retrieval with the use of the Protégé built-in query mechanism.

During the STMO ontology development, the knowledge concerning selected methods is identified, analyzed and placed into the knowledge base. The Examples 1-6 present the first group of methods introduced into the STMO knowledge base.

Example 1: The SecureMark System, ROSI-based, developed by SageSecure, NY, USA (Sonnenreich et al., 2006).

The SecureMark system is the result of the R&D work conducted at the SageSecure, aimed at the security benchmarking method, which gives the “consistently repeatable results that

are strongly correlated to financial performance”. The variants of the security expenditures are considered, using the Return on Security Investment calculations. The key ROSI parameter, the “Risk exposure”, is measured as the productivity loss caused by the security issues. The method is focused on solving everyday security incidents which constitute the significant amount of the aggregated loss. To better assess the security investment two additional well-known factors are used, i.e. NPV (Net Present Value) and IRR (Internal Rate of Return). The method concerns the information security issues and is “almost compliant” (ca. 95%) with ISO17799.

Example 2: The ontology-centred architecture for the technology risk management in the banking industry, developed at the University of Mannheim, Germany (Atkinson et al., 2006).

To foster the technology risk management in the banking sector, an ontological approach is proposed, which is compliant with the Basel II recommendation. The method, based on the ontological model, encompasses processes of the identification, measurement and controlling activities related to the risk self-assessment, loss database creation, model building and regulatory requirements. The ontology encompasses the risk-related issues (i.e. technology, process, human, activity, resource, effects, loss, gain, etc.) with respect to the banking activities (i.e. trading, credit, accounting activities, etc.) influencing the value chain. The identification process (the self-assessment) is questionnaire-based. The risk measurement uses the Monte Carlo simulation algorithms. During the risk controlling process the calculated risk is compared with the real data, including these retrieved from the loss database. The ontology is also used for the above mentioned processes integration and for reasoning. The method is supported by the Java-based tool, encompassing applications for three particular processes, using the common database. The tool is a combination of the enterprise resource planning (ERP) system and the knowledge management system. The former provides monetary data and makes calculations, the latter precisely describes the risk issues.

Example 3: The goal-oriented security trade-off modelling and analysis method, focused on the software development from the University of Toronto, Canada (Elahi & Yu, 2008).

During the software systems development the security or privacy issues should be taken into consideration in the same way as other kinds of requirements, e.g.: functionality, usability and performance. For the given project it is necessary to perform a trade-off analysis issuing the set of the requirements feasible to implement. The presented method is based on a goal-oriented framework for modelling and analyzing security trade-off, integrated with the design framework. The trade-off results can be taken into consideration during architecture design. The trade-off cases are sampled in the knowledge base tool. The first part of the framework is responsible for the security trade-off analyzing in a multi actor environment and with the use of the “i*” notation. This conceptual modelling technique considers three kinds of concepts: explicit goals, actors and security specific concepts (threats, vulnerabilities, risk, etc.). The security goals are usually about conflicts with design objectives derived from the stakeholders’ expectations (users, administrators, top managers, customers) and take into consideration other aspects, such as the non-functional requirements, standards, security policies, etc. The designers are the key actors and the other actors compete with them in achieving their particular goals. The second part of the framework is responsible for the requirement engineering issues, and the third is the knowledge base. The framework was used to model the NIST security guidelines and other

textbooks. Three validation cases were performed for the framework. The method was implemented in the identity management of an industrial application. The security trade-off analysis procedure and qualitative measurement scale are the key issues of the method.

Example 4: ATAM – Architecture Trade-off Analysis Method, developed in the Software Engineering Institute of the Carnegie Mellon University (CMU-SEI) USA, (Kazman et al., 2000).

Business processes rely on software architectures. They are very complex and their development needs deep analyses and many design trade-offs, particularly those which affect the achievement of quality attributes such as performance, availability, modifiability and security. The ATAM is an evaluation method to analyze if the considered architecture decisions satisfy particular quality goals related to the mentioned attributes and how these goals mutually interact and cause trade-off against each other.

The ATAM provides a framework to model quality attributes and decisions related to them. The introduced quality scenarios express the elementary quality attribute-related requirements, in other words – scenarios express goals. The achievement of these scenarios depends on the chosen tactic. The alternative tactics are allowed. The importance of the given scenario and the difficulties in achieving its goal is assessed by means of a three-level scale: High/Medium/Low. The ATAM is a multi-step process. The architectural approaches are identified after the presentation of the ATAM to the assembled stakeholders, the identification of the business drivers and the description of the proposed architecture (explaining how it addresses the business drivers). Next the “quality attribute utility tree” is generated, which presents “overall goodness” of the system. The system utility is described by the above mentioned quality attributes, including the security attribute. On this basis quality scenarios are specified with their features (stimuli and response, difficulties, importance, etc.) and prioritized. Then the architectural approaches are analyzed. The approaches addressing high-priority factors are analyzed and the architectural risks, sensitivity points, and trade-off points are identified. The risks concern decisions that have not been made during the architecture development or decisions with unclear consequences. The sensitivity points are parameters in the architecture to which some measurable quality attribute is highly correlated with other attributes. The trade-off points are understood as the architecture elements including more than one sensitivity point, where the measurable quality attributes are affected differently by changing that parameter (the increase of one attribute value causes the decrease of the other). Next steps encompass brainstorming, prioritizations of scenarios, and the architecture refinement and reporting with the stakeholders’ participation.

The ATAM method is supported by tools, elaborated by different developers. An example of such a tool is ArchKriti – a software architecture-based design and evaluation tool suite (Vallieswaran & Menezes, 2007) which supports the following activities in the architecture-based software development: quality attribute brainstorming, attribute-driven design, architecture evaluation and documentation. Another example is the ATAM Assistant (Lionberger & Zhang, 2007) that is a single application to manage, visualize, and report on all of the artefacts generated during the ATAM process.

Example 5: The integrated Security Verification and security solution Design Trade-off analysis (SVDT) (Houmb et al., 2006).

The work presents the model-driven development method, called SVDT, which is based on the UMLsec notation and the Bayesian Belief Network (BBN) concept. The UMLsec is an

UML extension, introduced by Jürjens, providing a unified approach to security features description during the secure systems development. Established formal rules of security engineering can be encapsulated and hence made available to a wider group of developers and evaluators. In the SVDT, the UMLsec and its supporting tool are used to specify security requirements (project goals) and to verify design alternatives against these requirements. The trade-off decision maker analyzes these alternatives using the ROSI calculation and chooses the best one with respect to the criteria. The trade-off concerns four issues: solution effect, solution cost, misuse frequency and misuse impact. The trade-off input parameters are: security risk acceptance, policies, standards, laws and regulations, priorities, business goals, budget and time to market. The ROSI calculation is based on the Bayesian Belief Network. The SVDT uses the Common Criteria standard for project evaluation on the static level.

Example 6: Mission Oriented Risk and Design Analysis (MORDA), focused on IT military applications, Department of Defense (DoD), USA, (Buckshaw et al., 2007).

MORDA is a top-level information assurance methodology used for critical systems in the military application domain. To evaluate the information system designs, MORDA integrates different approaches: the risk analysis techniques, multiple objective decision analysis models and portfolio analysis techniques. MORDA is a quantitative risk assessment and risk management process. The process helps to allocate the resources encompassed by the information system design and needed for the system operation. This allocation ensures that the information system is operable in a hostile and malicious environment. It has been applied in seven major Department of Defense risk assessment studies, including the assessment of the Global Command and Control System (GCCS). GCCS is a suite of applications residing on the Secret Internet Protocol Router Network, which is a collection of secret LANs connected to the backbone. The important issue for this system is to “determine the optimum of countermeasures to be applied to the system to provide information assurance”. The trade-off concerns the security related benefits on one hand and the cost of the applied countermeasures on the other hand. MORDA is based on attack tree models, qualitative/quantitative information assurance models and multiple objective decision analysis models, as well as the previously developed MORA (Mission Oriented Risk Analysis) of the Information Assurance Technical Framework (IATF). Into the mission processes the countermeasures are built in. The countermeasures are elaborated by the Socrates quantitative risk assessment model and tool using the output of “User Mission Support Needs”, which is the first of the mission oriented processes. MORDA uses a multiple objective decision analysis and helps to elaborate the adversary attack model (attack tree based), the user model (to maximize the network functionality, interoperability, bandwidth, and easy operation for end users) and system provider model (to maximize the control of evolution path and easy operation for end users and warfighters, to minimize resources and schedule slips, e.g. for testing, fielding, etc.). The considered countermeasure and design options (CDO), e.g. “Install Firewall”, “Install VPN”, “Auditing” are scored against the likelihood of detection, likelihood of success, adversary resources required and cost (including: the research, procurement, installation and maintenance). Using the Socrates risk analysis all combinations of possible to use CDOs are analyzed, and the best architecture is proposed. It allows to achieve the maximum benefits with respect to the existing constraints (portfolio optimization). The recommended architecture solutions are implemented in the system and maintained. The ontological representation of the MORDA

Searching within the results issued by other query is possible too, as well as storing the defined queries.

Fig. 3. The STMO simple query example using the Protégé query facilities (Protégé, 2008)

Let us suppose that the knowledge base user wants to find methods on the Security-Cost trade-off decision, implying IT-related measures and allowing the portfolio optimization. For the *STOMethod* class she/he defines three queries, shown in the Fig. 3. Six matching methods were found. It was done by mistake because the AND/OR condition was not set properly. The user changes the option to “Match All” and, at this moment, only one method, i.e. MORDA was found (not shown), satisfying the query condition.

Currently any queries, based on the existing slots can be defined. The usefulness of this tool will increase when more methods will be analyzed and placed into the knowledge base. Instead of the Protégé Query, a more enhanced query mechanism can be used, allowing to implement the enhanced competency questions.

5. Conclusions

The importance of the security trade-off issue is growing, especially for the organizations representing complex environments with a high risk exposure, having limited resources and complex relations between security and business processes. Usually these organizations need long-term planning to integrate costly and high consequence measures into their business processes. To have at one's disposal a right security trade-off method, supported by an effective tool, is the key issue for many today's organizations irrespective of their domain and character. The selection and use of the right method is difficult and requires specialized knowledge. On one hand, the security knowledge, including the security trade-

off related knowledge, is broadly available in different form (reports, books, standards, documentations, software embedded, guidelines) and from different sources (experts, technology providers, researchers, Internet, databases, standard bodies, security promoting organizations, etc.). On the other hand, however, this knowledge is hard to acquire because some terms are not commonly understandable, and sometimes it is hard to extract relevant pieces of knowledge for a given application.

The chapter presents the ontology-based framework for the acquisition and management of knowledge dealing with security trade-off methods, helping to select a proper method and tool for the given application with respect to existing limitations.

The STMO ontology is a prototype, validated on some examples of knowledge acquisition dealing with several methods. The validation can be concluded in the following way:

1. The assumed ontology domain is extremely wide and varied. Particular methods or tools are rather poorly described, usually with the use of terms convenient for their authors. There are no unified ways of presenting these methods, allowing to compare their features, possibilities, usefulness, etc. Generally, there is a shortage of information on the status of applications, performed case studies, allowing to evaluate their usefulness in practice. Generally, the methods and tools considered there need further and more exhaustive investigations, including the use of the particular methods in real environment (case studies).
2. Most of the identified methods are complex risk management methods. They are usually matured, broadly used by professionals and supported by developed tools. There are relatively few true security trade-off methods and, mostly, they are under development. These methods are poorly supported by tools. The security trade-off methods can be considered as the extension of the risk management methods.
3. Most of the security trade-off methods are mixture-mode methods, integrating different approaches, e.g. SVDT uses the UMLsec for the solution variants specification, ROSI scheme for risk assessment, BBN for risk calculation and for elaborating the trade-off decision. For this reason, the given method can belong to several categories.
4. Most methods are related to the information technologies. It seems obvious because these technologies support business processes in other application domains on one hand, and the risk issue is inherent to these technologies on the other hand.
5. The taxonomy of classes and the set of properties, specified at the beginning, were slightly modified during the ontology validation. A serious revision of STMO should be done after acquiring the knowledge on more methods, representing more application domains.

The positive issue is the STMO ontology ability to acquire considerable knowledge from the ontology domain with the use of the recognized tool (Protégé) and standards (OWL-DL). The negative issue is that only several methods and tools were placed into the knowledge base. The knowledge acquisition was much more difficult and laborious than it had been expected. The particular investigated methods and tools are described informally in different, and sometimes incomplete, ways, depending on the author's style and used terms, but the ontology requires uniform, concise knowledge representation.

There is a need to extend the STMO ontology and the related knowledge base by analyzing other methods and application domains, sampling related knowledge to achieve a critical mass of knowledge, allowing effective use of the knowledge base. The future investigation should be focused around some sources of the knowledge concerning the STMO ontology domain, for example:

- the book (Hilson et al, 2007), giving an exhaustive review of methods, focused on varied risk management (RM) issues: Strategic RM, Corporate governance, Financial RM, Business continuity and disaster recovery, Reputational RM, Risk-assessed marketing, Operational RM, Project risk management, Environmental RM, Legal and contract RM, Technical RM, Fraud RM, Counter-terrorism RM;
- the monograph (Graham et al, 2006), focusing on the methods supporting business continuity issues;
- the publication (COSO II, 2004), presenting in details corporate risk management methods and techniques;
- the book (Aven, 2008), reviewing and discussing in details different approaches and applications of risk management;
- the author's book (Bialas, 2007), reviewing different approaches of risk management, focused on IT applications;
- sources of knowledge about methods focused on non-IT application domains;
- project communities, e.g. EU 7th Framework Programme;
- the ENISA Inventory of Risk Management/Risk Assessment Methods and Tools (ENISA, 2009);
- expert communities, e.g. SARMA (Security Analysis and Risk Management Association) community (SARMA, 2009);
- the author's own R&D works focused on the "IBOTRAM – Integrated, Business-Oriented, Two-Stage Risk Analysis Method" and its implementation in business and public organizations; the method is ROI/ROSI-based and supports implementations of the Information Security Management Systems (ISMS), compliant with ISO/IEC 27001 (Bialas, 2007), (Bialas & Lisek, 2007);
- experiences gained by collaborating researchers and the knowledge from many other sources; please note that only (Hilson et al, 2007) enumerates about 20 organizations and 27 standards or best practices dealing with risk management.

Concurrently with the knowledge acquisition, it is necessary to perform some researches, including case studies, focused on the better knowledge understanding and structuring. This will allow to elaborate more enhanced competency questions. Different nuances should be considered, e.g.: between security investments and other kinds of investments (safety, business, defence), understanding of key terms (risk, measure, strategy, benefit, effectiveness, etc.) in different application domains.

Both main objectives of this work mentioned at the beginning of the chapter were fulfilled. The STMO prototype is able to support researchers in conceptualization and knowledge acquisition. The results of their investigations in the realm of security methods and tools can be expressed in a uniform and precise way, i.e. as a knowledge base. Researchers and other interested parties can retrieve and analyze this knowledge. The chapter presents a complete example of the ontology creation process using the basic knowledge engineering principles and the proven tool, open to general use.

6. Acknowledgments

This work was conducted using the Protégé resource, which is supported by the grant LM007885 from the United States National Library of Medicine.

7. References

- Atkinson, C.; Cuske, Ch. & Dickopp, T. (2006). Concepts for an Ontology-centric Technology Risk Management Architecture in the Banking Industry, *Proceedings of 10th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW'06)*, pp. 21-29, ISBN: 0-7695-2743-4, Hong Kong, October 2006, IEEE Computer Society
- Aven, T. (2008). *Risk Analysis. Assessing uncertainties beyond expected values and probabilities*, Wiley, ISBN: 978-0-470-51736-9, New Jersey
- Bialas, A. (2007). *Bezpieczeństwo informacji i usług w nowoczesnej instytucji i firmie*, Wydawnictwa Naukowo-Techniczne, ISBN 978-83-204-3343-2, Warszawa, (in Polish, English title: „Information and services security within modern organizations and companies” – book published by Scientific and Technical Publishing House, Warsaw)
- Bialas, A. & Lisek K. (2007). Integrated, business-oriented, two-stage risk analysis. *Journal of Information Assurance and Security*, Vol. 2, Issue 3, Atlanta, September 2007 (205-210), ISSN 1554-10, www.dynamicpublishers.com/JIAS
- Bialas, A. (2008). Ontology-based Approach to the Common Criteria Compliant IT Security Development, *Proceedings of the 2008 International Conference on Security and Management – SAM'08 (The World Congress In Applied Computing)*, Las Vegas, July 2008, CSREA Press
- Bialas, A. (2009). Ontology-based Security Problem Definition and Solution for the Common Criteria Compliant Development Process, *Proceedings of 2009 Fourth International Conference on Dependability of Computer Systems (DepCoS-RELCOMEX 2009)*, Brunow, Poland, June-July 2009, IEEE Computer Society, Los Alamitos, Washington, Tokyo, pp. 3-10, ISBN 978-0-7695-3674-3
- Buckshaw, D.L.; Parnell, G.S.; Unkenholz, W.L.; Parks, D.; Wallner, J.M. & Saydjari, O.S. (2005). Mission oriented risk and design analysis of critical information systems. *Military Operations Research*, Vol. 10, No. 2, pp. 19-38, ISSN 0275-5823
- Coras. (2009). *The CORAS Method*, <http://coras.sourceforge.net/index.html>, accessed March 2009
- COSO II. (2004). *Enterprise Risk Management – Integrated Framework: Executive Summary and Framework, Application Techniques*, Vol. I and II, COSO – The Committee of Sponsoring Organizations of the Treadway Commission (Polish edition, 2007)
- CSL – Computer Science Laboratory. Security ontologies in OWL. (2008), <http://www.csl.sri.com/users/denker/owl-sec/ontologies/>, accessed July 2008
- DAML Services – Security and privacy. (2008), <http://www.daml.org/services/owl-s/security.html>, accessed July 2008
- Ekelhart, A.; Fenz, S.; Goluch, G. & Weippl, E. (2007a). Ontological Mapping of Common Criteria's Security Assurance Requirements. In: *New Approaches for Security, Privacy and Trust in Complex Environments*, Venter, H; Eloff, M.; Labuschagne, L.; Eloff, J. &

- von Solms R., (Eds.), pp. 85-95, Springer, ISBN 978-0-387-72366-2, Boston, http://publik.tuwien.ac.at/files/pub-inf_4689.pdf, accessed March 2009
- Ekelhart, A.; Fenz, S.; Goluch, G.; Riedel, B. et al. (2007b). Information Security Fortification by Ontological Mapping of the ISO/IEC 27001 Standard, *Proceedings of the 13th Pacific Rim International Symposium on Dependable Computing*, pp. 381-388, ISBN 0-7695-3054-0, 2007, IEEE Computer Society, Washington DC, USA, http://publik.tuwien.ac.at/files/pub-inf_4689.pdf, accessed March 2009
- Ekelhart, A.; Fenz, S.; Klemen, M. & Weippl, E. (2007c). Security Ontologies: Improving Quantitative Risk Analysis. *Proceedings of the 40th Hawaii International Conference on System Sciences (HICSS'07)*, pp. 156-162, ISBN: 0-7695-2755-8, Big Island, Hawaii, 2007, IEEE Computer Society Press
- Elahi, G. & Yu, E. (2008). A Goal Oriented Approach for Modeling and Analyzing Security Trade-Offs, *Proceedings of the 26th International Conference on Conceptual Modeling*, pp. 375-390, ISBN 978-3-540-75562-3, Auckland, New Zealand, November 2007. Lecture Notes in Computer Science, Springer Berlin / Heidelberg
- ENISA Inventory of Risk Management/Risk Assessment Methods and Tools. (2009). http://www.enisa.europa.eu/rmra/rm_home.html accessed April 2009
- Graham, J.; Kaye, D. & Rothstein P.J. (Ed). (2006). *A Risk Management Approach to Business Continuity*, Rothstein Associates Inc., ISBN: 1-931332-36-3, Brookfield, Connecticut
- Hilson, D. (Ed). (2007). *The Risk Management Universe – a Guided Tour*, BSI Business Information (BIP 2036), ISBN: 978 0 580 50346, London
- Herzog's Security Ontology. (2006). Linköping University, available at (accessed July 2008): <http://www.ida.liu.se/~iislab/projects/secont/>
- Houmb, S.H.; Georg, G.; Jürjens, J. & France, R. (2006). An Integrated Security Verification and Security Solution Design Trade-off Analysis, available at (accessed April 2009): <http://mcs.open.ac.uk/jj2924/publications/papers/secse06.pdf>
- Kazman, R.; Klein, M. & Clements, P. (2000). ATAM: Method for Architecture Evaluation, Technical Report CMU/SEI-2000-TR-004 ESC-TR-2000-004, Carnegie Mellon University, Software Engineering Institute, August 2000
- Kim, A.; Luo, J. & Kang, M. (2005). Security Ontology for Annotating Resources, *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE, Proceedings part II*, pp. 1483-1499, Agia Napa, Cyprus, October - November 2005, LNCS Springer, ISBN 978-3-540-29738-3, Berlin / Heidelberg
- Lionberger, B. & Zhang, C. (2007). ATAM Assistant: A Semi-Automated Tool for the Architecture Tradeoff Analysis Method, *Proceedings of the Software Engineering and Applications (SEA 2007)*, Cambridge, USA, November 2007
- Lockstep. (2004). *A Guide for Government Agencies Calculating Return on Security Investment*, Government Chief Information Office (GCIO), version 2.0, Lockstep Consulting
- Martimiano, L.A.F. & Moreira, E.S. (2005a). Using ontologies to assist security management. *Proceedings of the 8th International Protégé Conference*, ISBN, Madrid, July 2005, <http://www.ppgia.pucpr.br/~maziero/pesquisa/ceseg/sbseg06/conteudo/artigos/resumos/19513.pdf>, accessed June 2008
- Martimiano, L.A.F. & Moreira, E.S. (2005b). An OWL-based Security Incident Ontology. In *Proceedings of the 8th International Protégé Conference*, ISBN, Madrid, July 2005, <http://protege.stanford.edu/conference/2005/submissions/posters/poster-martimiano.pdf>, accessed June 2008

- Noy, N.F. & McGuinness, D.L. (2001). *Ontology Development 101: A Guide to Creating Your First Ontology*, *Stanford Knowledge Syst. Lab. Tech. Rep. KSL-01-05 and Stanford Medical Informatics Tech. Rep. SMI-2001-0880*, Stanford University, CA. <http://www-ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html>, accessed May 2008.
- Protégé Ontology Editor and Knowledge Acquisition System, v.3.4. (2008). Stanford University. <http://protege.stanford.edu/>, accessed May 2008.
- REI Ontology Specifications, ver. 2.0. (2008). University of Maryland, available at (accessed May 2008): <http://www.cs.umbc.edu/~lkagal1/rei/>
- SARMA – Security Analysis and Risk Management Association. (2009), <http://sarma.org/>)
- Secure Business, Common Criteria ontology. (2008). available at (accessed October 2008): <http://research.securityresearch.at/research/focus/common-criteria-security-assurance/>
- Sonnenreich, W.; Albanese J. & Stout, B. (2006). Return On Security Investment (ROSI): A Practical Quantitative Model. *Journal of Research and Practice in Information Technology*, Vol. 38, No. 1, February 2006, pp. 55-66, ISSN 1443-458X
- Tsoumas, B.; Dritsas, S. & Gritzalis, D. (2005). An Ontology-Based Approach to Information Systems Security Management, *Proceedings of 3rd International Workshop on Mathematical Methods, Models, and Architectures for Computer Network Security, MMM-ACNS 2005*, pp. 151-164, ISBN 978-3-540-29113-8, St. Petersburg, Russia, September 2005, Springer Lecture Notes in Computer Science (LNCS), Volume 3685/2005, Berlin / Heidelberg
- Vallieswaran, V. & Menezes B. (2007). ArchKriti: A Software Architecture Based Design and Evaluation Tool Suite, *Proceedings of the International Conference on Information Technology ITNG '07*, pp. 701-706, ISBN:0-7695-2776-0, Las Vegas, USA, April 2007, IEEE Computer Society
- Vorobiev, A. & Bekmamedova, N. (2007). An Ontological Approach Applied to Information Security and Trust, *Proceedings of the 18th Australasian Conference on Information Systems*, Toowoomba, December 2007, available at (accessed September 2008): <http://www.acis2007.usq.edu.au/assets/papers/144.pdf>,
- Wiki/ontology (2009). [http://en.wikipedia.org/wiki/Ontology_\(computer_science\)](http://en.wikipedia.org/wiki/Ontology_(computer_science))
- Yau, S.S.; Yan, M. & Huang D. (2007). Design of Service-Based Systems with Adaptive Tradeoff Between Security and Service Delay, *Proceedings of the 4th International Conference on Autonomic and Trusted Computing (ATC)*, pp. 103-113, ISBN 978-3-540-73546-5, Hong Kong, China, July 11-13, 2007. Lecture Notes in Computer Science, Springer Berlin / Heidelberg
- Yavagal, D.S.; Lee, S.W.; Ahn, G.J. & Gandhi, R.A. (2005). Common Criteria Requirements Modeling and its Uses for Quality of Information Assurance (QoIA), *Proceedings of the 43rd Annual ACM Southeast Conference (ACMSE'05)* Vol. 2, pp. 130-135, ISBN:1-59593-059-0, Kennesaw State University Kennesaw, Georgia, ACM New York

Advances in Computer Science and IT

Edited by D M Akbar Hussain

ISBN 978-953-7619-51-0

Hard cover, 420 pages

Publisher InTech

Published online 01, December, 2009

Published in print edition December, 2009

The book presents some very interesting and excellent articles for this divergent title. The 22 chapters presented here cover core topics of computer science such as visualization of large databases, security, ontology, user interface, graphs, object oriented software developments, and on the engineering side filtering, motion dynamics, adaptive fuzzy logic, and hyper static mechanical systems. It also covers topics which are combination of computer science and engineering such as meta computing, future mobiles, colour image analysis, relative representation and recognition, and neural networks. The book will serve a unique purpose through these multi-disciplined topics to share different but interesting views on each of these topics.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Bialas Andrzej (2009). Security Trade-off — Ontological Approach, *Advances in Computer Science and IT*, D M Akbar Hussain (Ed.), ISBN: 978-953-7619-51-0, InTech, Available from:

<http://www.intechopen.com/books/advances-in-computer-science-and-it/security-trade-off-ontological-approach>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen