

Semantics and Knowledge-based System for Occupational Health Safety

Alexandra Galatescu¹, Adriana Alexandru¹, Corneliu Zaharia², Andrei Popovici²

¹ National Institute for R&D in Informatics,
8-10 Averescu Avenue, Bucharest, 011455, Romania,
agal@ici.ro, adriana@ici.ro

² Stefan Nicolau Institute of Virology,
285 Mihai Bravu Avenue, 030304, Bucharest, Romania
corneliu.zaharia@virology.ro, andrei_popovici@hotmail.com

Abstract: The paper presents the conceptual framework, the basic inference and some implementation results in a system (under development) for the online training on the potential risks and events that can occur during the execution of certain activities. This system is an application of ontologies in the occupational health domain. The training (in a requested context and complying with domain-specific rules) results from the automatic and semantics-based discovery of the prevention documents, actions, methods etc that fit the user's request.

Keywords: occupational health; risk prevention; formal ontologies; ontology-based modeling; e-Training.

1. Introduction

According to the Joint ILO/WHO committee in Occupational Health (1950), the goal of the *occupational health* is the promotion and maintenance of the highest degree of physical, mental and social well-being of the workers in all domains, their protection from risks caused by their working conditions, negligence, lack of training etc [1]. Hence, the *occupational risks* are a special type of risks that appear in the work environments with a high probability of harming people or machines. Despite its importance in the risk management (RM) inside and across organizations, the occupational risk *prevention* is still a manual process, specific to domains like health, construction, transportation, industry, biology, etc. A proactive approach in RM relies on the risk early recognition and prevention.

Nowadays, the occupational risk prevention and management comply with the principles and methodology of the risk management process, a key process within both the private and public organizations [2].

The training in occupational risk prevention should advise the operator/ worker on the health, safety, security and environmental issues related to his work. He can ask for training before or during the execution of an activity or before the use of a certain machine.

The system described in this paper relies on the standard terminology proposed with ISO

CD31000 [2], [3], combined with the terminology common to several upper-level ontologies and process models.

There are several risk-related standards published by ISO and other standards bodies, as well as many proposals and principles that refer to risk management. In 2005, ISO has initiated a working group to develop a guidance standard on RM, ISO CD31000. In conjunction with this standard, the group has updated the ISO/IEC Guide 73-Risk Management – Vocabulary [4], that gives a basic vocabulary and the definitions of the RM generic terms. It encourages a mutual and consistent understanding and a coherent approach to the description of the RM activities.

In Europe, the risk prevention is subject of two directives Seveso I and Seveso II [5] that establish the domain terminology, the obligations and normative documents regarding the large scale industrial hazards.

In practice, there are products for the risk control in industrial environments and domain-specific standards and software tools for RM in health, environment, insurance, finances, construction, transportation, etc. Risk prevention is automated for the security of computers, Web, networks. Ontologies are also used mainly for the security management (of assets, networks, information systems, databases, etc). Some examples are in [6]-[10]. There is no system based on knowledge and semantics for risk prevention and for training and dynamic discovery of prevention information, documents and actions.

However, [11] proposes the risk evaluation and analysis along the life cycle of the construction projects, based on ontologies and on a conceptual model. They rely on a simpler reference ontology and model and have a different inference goal. Also, [12] gives an example of an ontology in OWL (Web Ontology Language) [13] for occupational health. And, [14] confirms the idea that a model of occupational risks is needed to describe relevant data in the context of event occurring and this data can be transformed into knowledge navigated using an intelligent search engine (similarly to the goal of the system presented in this paper).

Section 2 describes the ontology-based model proposed for risk prevention. Section 3 presents the basic inference that will be implemented in the system.

2. An Ontology-based Model for Occupational Risk Prevention

Figure 1 gives the general architecture of the system for risk prevention.

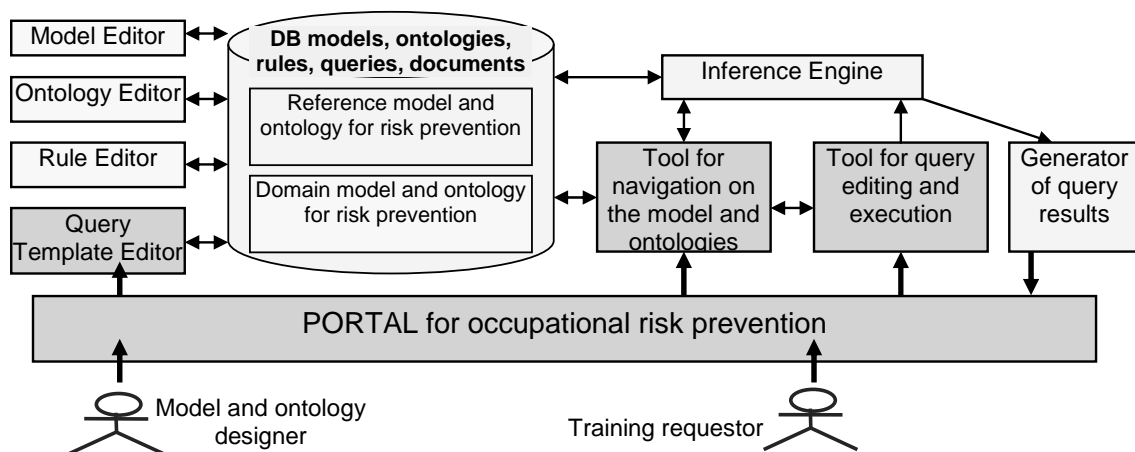


Figure 1. General architecture of the system for risk prevention (platform for risk design (left); and platform for risk evaluation and decision making (right))

The intended system will have components distributed on two platforms: *platform for the risk design*, i.e., for the risk identification, description and analysis (Figure 1, left); and *platform for the risk evaluation and decision-making* on the training query results (Figure 1, right). The two platforms share the repository composed of ontologies, rules, queries and documents.

The *platform for the risk design* contains: (1) a *Model Editor* to build and instantiate domain

or application specific models (e.g., the model for risk prevention in Figure 2 and exemplified in Figure 3); (2) an *Ontology Editor* to build and instantiate (specialization, composition or list-like) ontologies involved in models; (3) a *Rule Editor* to define or customize domain or application specific rules in an organization (e.g., for risk prevention); (4) a *Query Template Editor* to predefine or customize templates for application specific queries on a model (for example, queries for training in a certain domain).

A specific feature of this platform is that the rule and query editors are tightly integrated with the model and ontology editors (as exemplified for queries in Figure 4, where the concepts and attributes are directly seen and selected during the query composition)

The platform for the risk evaluation and decision-making contains: (1) a Tool for query editing and execution to dynamically compose the training queries and ask for their execution; (2) a Tool for the ontology and model navigation in order to compose a rule or query; (3) an Inference Engine, automatically

called after the submission of the query, in order to perform the automatic discovery of the query results (e.g., training documents and the appropriate prevention actions, previously registered in the system and referred to in the domain model). Besides the conditions and constraints in the query, the discovery will also rely on rules previously defined by the risk designer using the rule editor; (4) a Generator of the query results, called by the inference engine, after the result discovery, in

order to arrange the query results, according to the model and user's preferences.

From the conceptual point of view, the system integrates *three layers* representing the occupational risks and the context for their occurrence and prevention: semantic, modeling and execution layers.

The *semantic layer* is composed of the *reference ontology* and of *domain ontologies* that give the basic vocabularies for domains with potential risks. The domain ontologies are populated by domain experts (risk designers) using the ontology editor. They are represented by: (1) domain-specific *taxonomies*, i.e., hierarchies composed of concepts connected, in this system, by relationships like: *specialization or synonymy* or *composition* (part-of) or *list-like* relationships; and (2) *attributes* of and *constraints* upon the concepts and relationships in ontologies.

The concept attributes in any ontology can refer to external ontologies. E.g., the "domain" attribute of an "activity" in the Activity ontology can be selected from the Domain ontology.

list-like ontologies it is implemented only for ontology attributes, implicitly inherited by all concepts in that ontology.

The *modeling layer* is needed in addition to the semantic layer in order to represent *ontology-based models* for applications. Hence, a model is seen here as a union of relationships between concepts in different ontologies, along with their attributes and the constraints on them. In this system, the *inter-ontology relationships* are defined according to the reference model represented in Figure 2.

The *execution (technological) layer* represents the ontologies and models, the documents, rules, constraints and queries in a format interpretable by the software.

The concept types connected as in the reference model in Figure 2 and described in Table 1 root ontologies based on specialization, composition or list-like relationships. These ontologies have been proposed to help for the identification and classification of the risk factors, of the consequences and preventive measures, of the dangerous activities and of the processes they compose, etc.

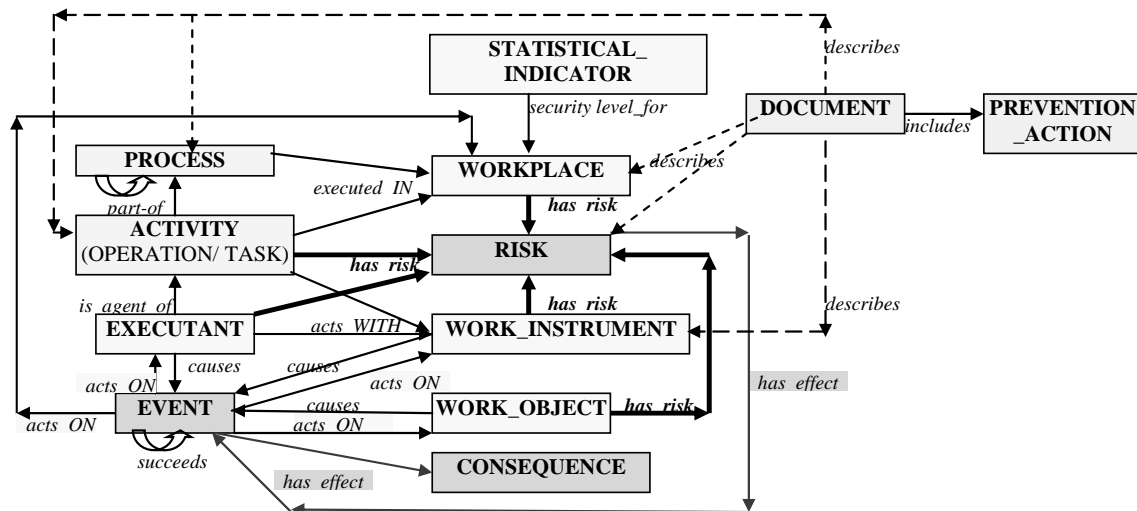


Figure 2. Basic types of ontologies and relationships in the reference model for occupational risk prevention

In this system, the ontology editor treats separately the specialization, composition and list-like ontologies, because each type of ontology has its specific features. E.g., the concept *attribute inheritance* is implicit only in the specialization ontologies. For the composition ontologies, it can be explicitly requested by the user for attributes of certain concepts. For

Figure 2 shows how the semantic and modeling layers for risk prevention are integrated from the conceptual point of view. The modeling layer represents the relationships between the ontologies defined on the semantic layer. These relationships have been selected depending on the needed reasoning on them and on the context for the

risk identification, analysis, evaluation and prevention, identified at this moment. The model can be dynamically enhanced with new ontologies, relationships, attributes and constraints that will be used in future rules, queries and inferences.

- A domain model (*application modeling content*) that instantiate the reference model. Domain models can be populated by the application designer and by another domain expert.

Table 1. Basic types of ontologies on the semantic layer for occupational risk prevention

<p>Activity (operation/ task): atomic operation executed independently or during a process inside the organization</p> <p>Consequence: outcome of an event or change in circumstances affecting the achievement of objectives [4]. An event may lead to a range of consequences. A consequence can have positive or negative effects. For the occupational risks, only the negative effects are considered.</p> <p>Document: a document containing prevention/ protection/ control instructions, regulations, rules or measures for risk prevention.</p> <p>Event: occurrence or existence of a particular set of circumstances. An unpredictable event is called "incident" [4]. It can be the consequence of the executants' action using a certain instrument and acting on a certain object.</p> <p>Executant (or Starter or Operator): the (human or material) agent that, during an activity, can cause unexpected events and also can be injured by them or can get professional diseases.</p>	<p>Prevention_Action: management action preventing the unexpected events or diseases. An example is the training of the operators in workplaces with potential risks.</p> <p>Process: a sequence of activities/ operations/ tasks in a certain domain and workplace. The activities in a process can be executed by different executants, at different moments and in different places.</p> <p>Risk: combination of an event probability and its consequences [4]. The event can take place in a certain workplace, during a certain activity/ task or resulting from a material source action (e.g., water, a substance, gas, etc).</p> <p>Work_Instrument: tool/ machine/ substance/ etc., used by the operator during an activity/ task. It can determine an event or be damaged by it.</p> <p>Work_Object: object existing at a workplace. It can determine an event or an event may impact on it. It can be material (e.g., a substance) or human (e.g., an infected patient in a hospital).</p> <p>Workplace: location in the organization where unexpected events can occur and affect/ destroy it.</p>
--	--

The modeling for risk prevention and control is today mainly a mathematical modeling complemented with formal methods to assess or measure the risks and to help the decision-making for their prevention. Also, this modeling is usually a domain-specific one for health/ financial/ insurance/ economic/ business/ etc. risks.

In the system described in this paper, the modeling layer is composed of:

- A reference model (*application modeling schema*), built by the application designer;

On the modeling layer, several reference models can be defined. A reference model can involve new ontologies or ontologies previously defined for other models.

The model is seen as an *ontology-based extension of an ER (Entity-Relationship) model* and graphs. There are three *main reasons* for the choice of the ER as underlying model:

- The diagrams / graphs are explicit and simple, dedicated to the domain experts;

- The information is represented similarly to the way of thinking, without unnecessary information;
- The diagrams / graphs are flexible: nodes and relationships can be dynamically added or removed, without code modifications.

The model in our system replaces the entity types with ontologies. This modification introduces an additional problem caused by the *semantic relationships* inside ontologies, in addition to the *modeling relationships* between entity types in the ER model.

The model graph definition in this system is summarized below.

ModelGraph = (ModelNode, ModelEdge), where

ModelNode = ConceptNode \cup
ConceptInstanceNode

Nodes can be concepts or concept instances.

ConceptNode = $\{(C_j(O_i), \{Subtree(C_j(O_i))\})\};$
 $\{CAtr_k(C_j(O_i), \{Subtree(C_j(O_i))\})\};$
 $\{v_{ijk}\}_{i=1:I, j=1:J, k=1:K}$ where

- O_i is an ontology in the model;
- $C_j(O_i)$ is a concept in O_i ;
- $Subtree(C_j(O_i))$ is the set of the semantic subtrees for the concept $C_j(O_i)$ (composed of the concept (specialization) subtypes or (composition) parts and of its synonyms). It is NULL, if the concept belongs to a list-like ontology and does not have synonyms.
- $CAtr_k$ is the set of attributes for a concept $C_j(O_i)$ or for a concept in one of sub-trees;
- v_{ijk} is the value of an attribute in the set $CAtr_k$.

So, a concept node is a concept in ontology, together with concepts in its specialization/ composition and synonyms sub-trees in the respective ontology. Each concept has attached values of its attributes.

ConceptInstanceNode = $\{(CIns_h(C_j(O_i),$
 $\{Subtree(C_j(O_i))\})\}; \{CInsAtr_k(CIns_h(C_j(O_i),$
 $\{Subtree(C_j(O_i))\})\}; \{v_{ijk}\}_{i=1:I, j=1:J, h=1:H; k=1:K}$
where

- $CIns_h$ is the set of instances of the concept $C_j(O_i)$ or of a concept in its semantic subtrees;
- $CInsAtr_k$ is the set of attributes for the concept $C_j(O_i)$ or for a concept in its semantic sub-trees, implicitly inherited by its instances. The attribute values inherited from the concepts can be changed afterwards.

So, a concept instance node is an instance of a concept along with the instances of all concepts in its specialization/ composition and synonyms sub-trees in ontology. The attributes of the concepts and instances have specific values attached to them.

Edges correlate the concept nodes or concept instance nodes, as follows:

ModelEdge = Inter-ConceptEdge \cup
Inter-ConceptInstanceEdge

where

Inter-ConceptEdge =
 $\{\text{RelConc}_j(\text{RelOnt}_{i_1, i_2, f}(O_{i_1}, O_{i_2}), CO_{i_1}, CO_{i_2});$
 $\{\text{RelAtr}_g(\text{RelOnt}_{i_1, i_2, f}(O_{i_1}, O_{i_2}))\};$
 $\{v_{i_1, i_2, f, g}\}_{i_1=1:I, i_2=1:I, j=1:J, f=1:F, g=1:G}$

ConcInsEdge =
 $\{\{\text{RelConcIns}_k(\text{RelConc}_j(\text{RelOnt}_{i_1, i_2, f}(O_{i_1},$
 $O_{i_2}), CO_{i_1}, CO_{i_2}), CIns_{O_{i_1}}, CIns_{O_{i_2}})\};$
 $\{\text{RelAtr}_h(\text{RelOnt}_{i_1, i_2, f}(O_{i_1}, O_{i_2}))\};$
 $\{v_{i_1, i_2, f, h}\}_{i_1=1:I, i_2=1:I, j=1:J, f=1:F, h=1:H}$

where:

- $\text{RelOnt}_{i_1, i_2, f}(O_{i_1}, O_{i_2})$ is a relationship type between two ontologies O_{i_1} and O_{i_2} ;
- CO_{i_1}, CO_{i_2} are concepts in the ontologies O_{i_1} and O_{i_2} ;
- RelConc_j is a binary relationship between two concepts CO_{i_1}, CO_{i_2} in the ontologies O_{i_1} and O_{i_2} ;
- RelAtr is the set of attributes specific to the relationship type $\text{RelOnt}_{i_1, i_2, f}$ between the ontologies O_{i_1} and O_{i_2} . These attributes and their values are inherited by the relationships between concepts in the two ontologies and by the relationship between their instances. The initial values of the attributes (given at the definition of the relationship type) can be changed for

the relationships between concepts and between instances.

- $v_{i1, i2, f, h}$ is the value of an attribute $RelAtr_h$;
- $CInsO_{i1}$ and $CInsO_{i2}$ are instances of the concepts CO_{i1} and CO_{i2}
- $RelConcIns_k$ is a relationship between the instances of the concepts CO_{i1} and CO_{i2} (previously correlated by the relationship $RelConc_j$).

The edge between two concepts CO_{i1} , CO_{i2} or between their instances has the name of the generic relationship type and is described by its attributes, possibly with different values.

The *properties of the model graph* are:

- It has the *order* (the number of nodes) maximum $\sum_{i=1:I} (\text{Number of concepts}(O_i))$;
- It has the *size* (the number of edges) maximum
- No of concepts(O_1) x No of concepts(O_2) x...x No of concepts(O_I)
- It has the *degree of an edge* (number of connected concepts) always 2.
- It has the *degree of a node* (the number of edges that connect to the node) maximum I-1 (the number of ontologies, excepting the ontology the node belongs to).
- It is a *directed graph*, because each edge is directed from a source node (selected from a source ontology) to a destination node (selected from a destination ontology). Some edges can be *bidirectional*.
- It is a *labeled graph* because both nodes and edges have names (*concept name* or *relationship name*);
- It can be seen as a *set of bipartite graphs* because each relationship correlates concepts selected from two disjoint ontologies;
- It is a *weighted graph* because each edge (relationship) has attributes with values;
- It can be an *incomplete graph*, because:

- certain ontologies might not be connected to each other by relationship types, during the model definition;
- if there is a relationship defined between two ontologies O_{i1} and O_{i2} , certain concepts in O_{i1} might not be correlated with concepts in O_{i2} , at certain points in time.
- It can contain *cycle sub-graphs* of length n (at least 3) when for the nodes c_1, \dots, c_n , the edges are $c_{i-1}c_i$ for each $i = 2, \dots, n$ and we have an edge $c_n c_1$.
- It *cannot have a loop edge* (that connects to a node at both ends) because each edge connects concepts or concept instances in disjoint ontologies.

The *benefits from an ontology-based model* in general and, in particular, for risk prevention are briefly enumerated below. First, the types of concepts in ontologies and the relationships between them in the model, as well as the reasoning on them, are explicit (external to the application code) and independent of the application tools. Second, the ontologies can be shared by different diagrams or models (e.g., for risk monitoring and control, in addition to risk prevention). Third, the separation of the model from the ontology content makes the model flexible, adaptable and extensible. The tools for ontology editing and navigation may differ from the tools for the model editing and navigation. Also, the reasoning on the model can be separately implemented from the reasoning on ontologies.

The basic relationships in the reference model in Figure 2 are described in Table 2.

The generic concepts, their attributes, relationships and constraints in the reference model are specialized and instantiated by the risk designers, resulting in *domain models* (e.g., for biological or industrial risks). For any domain concept, the designer instantiates the concept attributes defined in the ontology. They can be reference attributes, (for the concept unique identification) or inherited attributes (that can be ontology or concept specific attributes). Also, by their instantiation, the inter-ontology relationships in the reference model become domain-specific relationships between concrete concepts in ontologies or between concept instances.

Figure 3 exemplifies some ontologies, relationships and attributes that compose the risk prevention model.

The system is developed using Microsoft Visual Studio 2008 and about Suite product

for ASP.Net [19]. It integrates the expression evaluator given in [15], adapted to the use of concepts and attributes in ontologies; and, the interface for the rule and query editors is inspired from [16].

Table 2. Basic relationships in the reference model for occupational risk prevention

<p>Activity->Process is a “<i>part-of</i>” relationship between activities and the process they belong to. In a process, the activities might be executed by operators in different departments and even in different organizations. The risks should be tracked for each activity, but also for each process in/cross organizations.</p> <p>Activity->Workplace and Process->Workplace relationship “<i>executed_IN</i>” is necessary to track the risks per activity, process and workplace at the same time.</p> <p>Event->Consequence relationship “<i>has_effect</i>” associates the events with their consequences.</p> <p>Event->Workplace are correlated by the relationship “<i>acts_ON</i>” in order to associate the events to the workplaces they can damage.</p> <p>Document->Risk relationship “<i>describes</i>” associates to the identified risks the documents (and the actions the documents refer to) necessary for the risk prevention.</p> <p>Executant->Activity relationship “<i>is_agent_of</i>” and also Activity->Work_Instrument and Executant->Work_Instrument relationship “<i>acts_WITH</i>” are necessary for reasoning on an operator-activity-machine sub-model.</p>	<p>Executant->Event relationship “<i>causes</i>” helps for the identification of the events that an operator might determine by his work. The inverse relationship “<i>acts_ON</i>” between Event->Executant helps for the identification of the executants that can be injured after certain events.</p> <p>Process->Process relationship is a “<i>part-of</i>” relationship between a process and its sub-processes with potential risks that should be tracked.</p> <p>Risk->Event relationship “<i>has_effect</i>” associates the identified risks to the events they may produce.</p> <p>The reference model in Figure 2 also associates the elements with potential risks (types of concepts like Activity, Executant, Workplace, Work_Instrument, Work_Object) with their specific risks, by the relationship “<i>has_risk</i>”.</p> <p>Work_Instrument->Event relationship “<i>causes</i>” is necessary to identify the events determined by the inappropriate use of a certain instrument. The inverse relationship “<i>acts_ON</i>” between Event->Work_Instrument is necessary to identify the instruments that can be damaged after certain events.</p> <p>Besides the executants and the work instruments, the unexpected events or diseases might be caused by other objects existing at the workplace. These events can be found by the relationship “<i>causes</i>” between Work_Object->Event. Also, the objects damaged by certain events can be found using the relationship “<i>acts_ON</i>” between Event->Work_Object.</p>
--	--

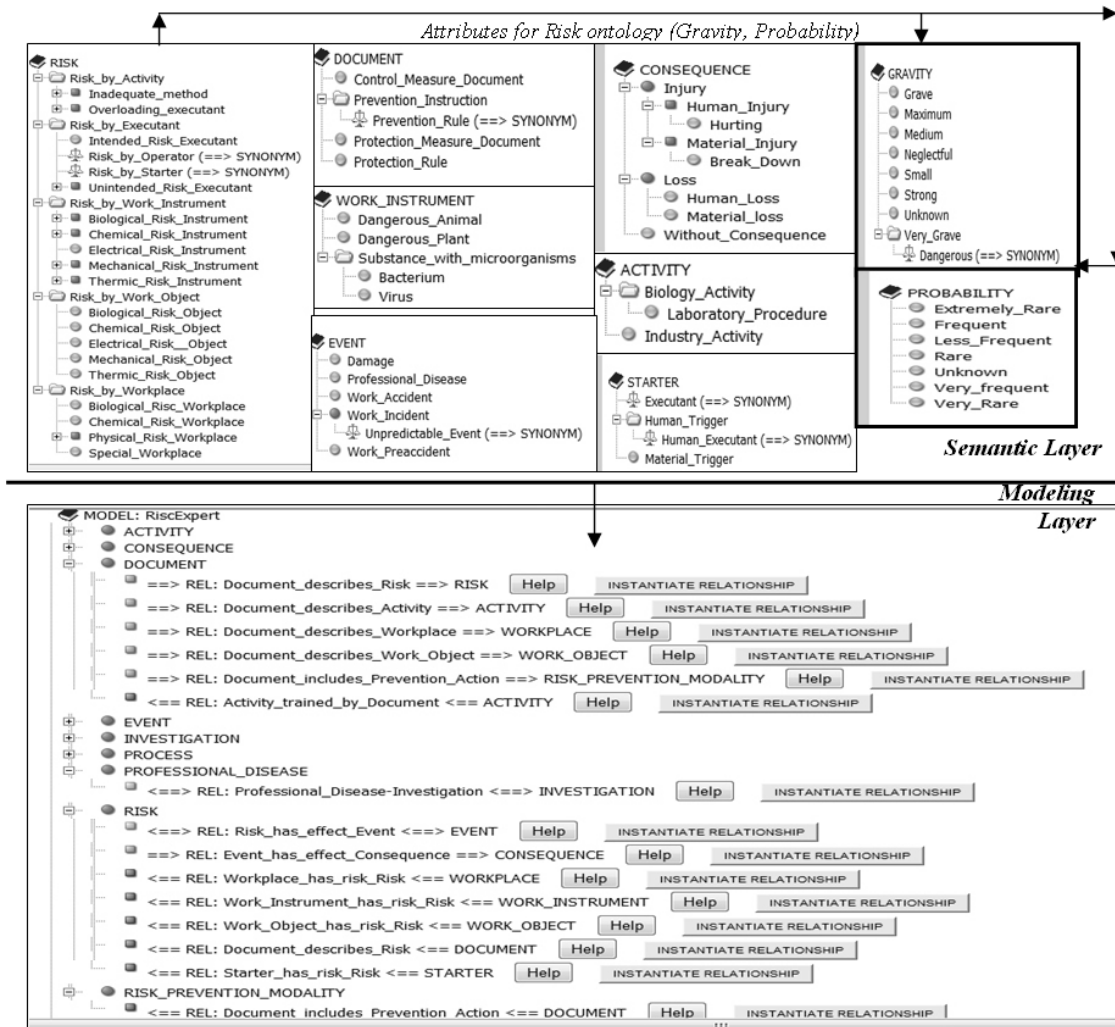


Figure 3. Examples of ontologies, relationships and attributes in the risk prevention model

3. Ontology-based Inference on the Model for Risk Prevention

The basic *knowledge and inference* for risk prevention will include (see also details in the case study given in [17] and in the example for the query composition in [18]):

- *Meta-reasoning rules* (predefined and encoded) used by the system for managing ontologies, models, deductive or querying rules. Meta-reasoning is implemented in the editors for ontologies, models, rules and queries, in the ontology or model navigators, in the expression parsers, in the inference engine.
- *Deductive rules* (user-defined, built using the *rule editor*) for inference on ontologies and models. New facts on ontologies and models (expressed in

THEN or ELSE consequent (conclusion)) are derived from existing (known) facts expressed in (IF) CONDITION (antecedent/ premise). These rules comply with the system's *rule model*.

- *Querying rules* (user-defined, built using a *query editor*) for the composition of complex queries upon ontologies and models. The querying rules are used in combination with the deductive rules. They comply with the system's *query model*.

Meta-reasoning rules on ontologies. These rules are predefined and automatically checked for the *ontology correctness*. They are used for:

- Adding an ontology to a model;
- Adding an ontology-specific attribute;

- Setting up an inter-concept relationship in ontology;
- Checking the attribute value compliance with the (ontology or concept) attribute definition: mandatory or optional existence of the attribute value; attribute value inheritance from a concept to its children; data type for the attribute value.
- Ontology attribute inheritance;
- Ontology navigation;
- Definition of a concept in ontology;
- Adding a concept specific attribute;
- Concept attribute inheritance;
- Adding a concept instance.

For the specialization ontologies, the concept attributes are automatically inherited from the parent concepts. For the composition ontologies, they are inherited only on the user's demand. The *values of the attributes* are also *inherited on demand*, excepting the values for the identification attributes (e.g., ID, concept author, concept creation date, etc) that cannot be inherited.

Meta-reasoning rules on relationships.

These rules are predefined and automatically checked for the *model correctness*. They are applied for:

- Adding an inter-ontology relationship to a model;
- Adding an inter-concept relationship. It instantiates an inter-ontology relationship;
- Adding a relationship between concept instances. It instantiates an inter-concept relationship;
- Adding an attribute to a relationship;
- Checking the mandatory existence and the cardinality of a relationship;
- Definition of the relationship direction.

Definition of deductive rules on ontologies and models. The definition below (in EBNF notation) summarizes the basic components of a deductive rule in this system.

Deductive Rule := CONDITION THEN ELSE;

This definition is similar to:

Antecedent ? THEN-Consequent:
ELSE-Consequent; or with
Premise THEN- Conclusion:
ELSE- Conclusion

CONDITION := Expression
THEN := Expression | NULL ;
ELSE := Expression | NULL ;
Expression := ([B-G_Operator]
Relational_Expression |
[Boolean_Expression])
(sequence-operator Expression) *) *
Relational_Expression :=
Operand relational_operator Operand
Boolean_Expression :=
[Operand] boolean_operator Operand |
Operand = boolean_value
B-G_Operator := boolean_operator |
grouping_operator
sequence-operator := space/ blank
boolean_operator := AND | OR | NOT
grouping_operator := (| [|]
relational_operator := = | > | < | <= | >=
boolean_value := True | False
Operand := Identifier | (Constant Expression) |
(Expression) | NULL
Constant_Expression :=
(Arithmetic_Expression)* |
String_Expression* | Constant
Arithmetic_Expression :=
[Constant] arithmetic_operator [Constant]
String_Expression := string_constant
[string_operator string_constant]
Constant := string | number
number_punctuation_symbol number
Identifier := Ontological_Identifier |
Modeling_Identifier | User_Concept |
(User_Concept relational_operator
Constant_Expression)
Ontological_Identifier := **Ontology_Concept**
|
(Ontology_Concept = **Concept_Instance**) |
(**Ontology_Reference_Attribute**
relational_operator Constant_Expression) |
(**Ontology_specific_Attribute**
relational_operator Constant_Expression) |
(**Concept_specific_Attribute**
relational_operator Constant_Expression)
Modeling_Identifier :=
(**Relationship_Reference_Attribute**

relational_operator Constant_Expression)
 | (Relationship_specific_Attribute
relational_operator
 Constant_Expression)

number := {numeric_character}-

number_punctuation_symbol := . | ,

arithmetic_operator := + | - | * | / | %

string_constant := "string" | 'string'

string_operator := &

string = {alphanumeric character }-

Rule evaluation, resulting in a TRUE or FALSE condition, depends on the instances (values) of the operands in the condition expression.

In CONDITION, THEN and ELSE, the relational expressions are mandatory for the rule execution. When an identifier is a concept instance it is necessarily assigned as value of a concept in ontology. The attributes are necessarily followed by a relational operator and a value.

Querying rules. A query is composed by the training requestor using the query editor. The query execution means the search on the model and ontologies for concepts and instances complying with the search condition and restrictions. The query has three parts (search query, associative search condition, descriptive search restrictions) and the following composition rules (see an example in Figure 4):

Search query (What information is needed for training, i.e., what types of concepts to search for in the model and ontologies?) is a Boolean expression with:

- *operands* that can be: *names of ontologies*; or *names of concepts in ontologies*. E.g., the user asks for the prevention rules and the prevention measures to be selected from the ontology Document and for the physical risks at the workplace to be selected from the Risk ontology).

- *Operators*: only Boolean and grouping operators.

Expected results from the query execution are:

- For each *name of ontology* in the Search Query, the system will find:

- All *concepts* in the specified ontology that are connected (directly or indirectly) with concepts in Search Condition, along with their *subtypes and synonyms*;
- All *instances* of the previously discovered concepts;
- For each *name of concept* in Search Query, the system will find:
 - The *subtypes and synonyms* of the specified concept that are connected (directly or indirectly) with concepts in Search condition;
 - All *instances* of the specified or discovered concepts (subtypes and synonyms).

Associative Search Condition (The *associative restrictions* (or associative context) for search, i.e., with what concepts in the model should be correlated the searched concepts?) is an expression with:

- *operands* that can be *concepts* in ontologies that can have as values *concept instances*.
- *operators*: "=", between a concept and its instance, Boolean, grouping and sequence operators.

E.g., the searched concepts should be correlated with the activity "Laboratory Procedure" selected from Activity ontology and with the work instrument "Substance with micro-organisms" selected from Work Instrument ontology. Or, they should be correlated with a particular procedure and a particular substance with micro-organisms assigned as instances of the concepts "Laboratory Procedure" and "Substance with micro-organisms".

Descriptive Search Restrictions (The *descriptive restrictions* (or descriptive context) for search, i.e., what values should have the attributes of the searched concepts or the attributes of the relationships they are involved in ?) is an expression with:

- *operands* that can be *concept attributes*, mandatorily with constants as values. The attribute value is in a certain relation (=, <,>, <=, >=, <>) with the respective attribute. A value can be an arithmetic expression of numeric constants or a string expression. Examples of operands

assigned with values are Gravity="High" or Probability="Frequent".

- *operators* that can be relational, arithmetic, Boolean, grouping, string and sequence operators.

After the query submission, the query is analyzed and semantically completed, the expressions for the associative condition and descriptive constraints are syntactically and semantically analyzed and, then, the search algorithm is executed.

Query definition in EBNF is as follows:

```

Defined_Query := Search_Query
                Search_Associative_Condition
                Descriptive_Restrictions
Search_Query := Query_Expression
Search_Associative_Condition :=
    Condition_Expression | NULL
Descriptive_Restrictions :=
    Restrictions_Expression | NULL
Query_Expression := [B-G_Operator]
                  Query_Operand |
                  Query_Boolean_Expression
                  (sequence-operator Query_Expression)*
Condition_Expression :=
    [B-G_Operator] (Condition_Operand
    | Condition_Relational_Expression
    | Condition_Boolean_Expression
    (sequence-operator Condition_Expression)*
Restrictions_Expression := [BG_Operator]
    (Restrictions_Relational_Expression |
    Restrictions_Boolean_Expression)
    (sequence-operator Restrictions_Expression)*
Query_Boolean_Expression :=
    [Query_Operand] boolean_operator
    Query_Operand
Query_Operand := Ontology_Name | Concept
Condition_Relational_Expression :=
    (Concept = Concept_Instance)
Condition_Boolean_Expression :=
    [Condition_Operand] boolean_operator
    [Condition_Operand]
Condition_Operand := Concept |
    (Condition_Expression)
Concept := Ontology_Concept
Restrictions_Relational_Expression :=
    Restrictions_Operand
    relational_operator Constant_Expression
Restrictions_Boolean_Expression :=

```

```

[(Restrictions_Operand |
  Restrictions_Operand = boolean_value)]
boolean_operator
[(Restrictions_Operand |
  Restrictions_Operand = boolean_value)]

```

```

Restrictions_Operand :=
Restrictions_Identifier
    | (Restrictions_Expression) | NULL
RESTRICTIONS_Identifier :=
    Ontological_Identifier |
    Modeling_Identifier
Ontological_Identifier :=
    (Ontology_Reference_Attribute
    relational_operator Constant_Expression) |
    (Ontology_specific_Attribute
    relational_operator Constant_Expression) |
    (Concept_specific_Attribute
    relational_operator Constant_Expression)
Modeling_Identifier :=
    (Relationship_Reference_Attribute
    relational_operator Constant_Expression) |
    (Relationship_specific_Attribute
    relational_operator Constant_Expression)

```

B-G_Operator, *sequence operator*, *Boolean operator*, *grouping operator*, *relational operator*, *Boolean value*, *string constant*, *string operator*, *string*, *Constant Expression*, *Arithmetic Expression*, *String Expression*, *Constant*, *arithmetic operator*, *number*, *number punctuation symbol* are defined like in the rule definition above.

Steps of algorithm for search on model.

The algorithm is divided into six basic steps (sub-algorithms):

1. *Analyze the Search Query expression* for its syntactic and semantic correctness:
 - 1.1 Check if the query expression is composed of operands that are only names of ontologies or concepts in ontologies;
 - 1.2 Check if the query expression contains only Boolean or grouping operators (AND, OR, NOT, (,)).
2. *Analyze the expression for the Associative Condition* in search, for its syntactic and semantic correctness:
 - 2.1 Check if the expression contains operands that are only concepts in ontologies, possibly assigned (by "=" operator) with names of concept instances;

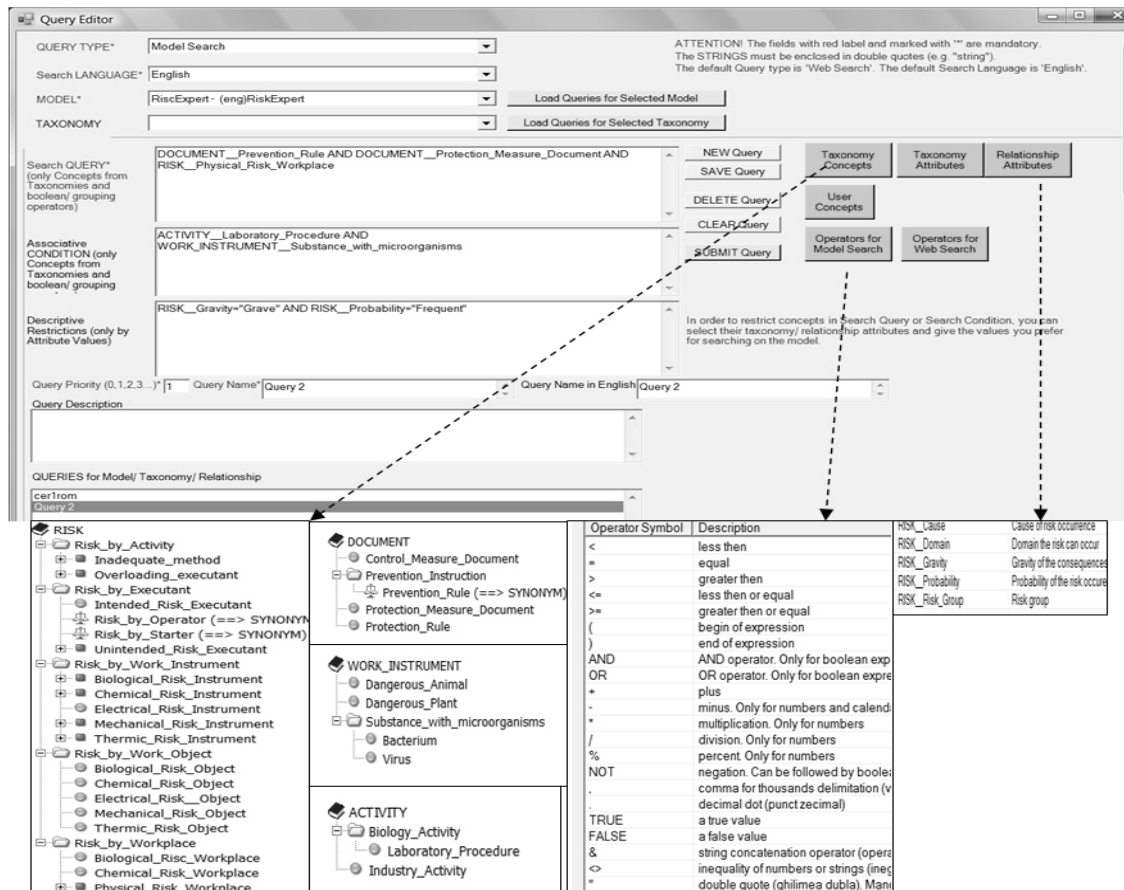


Figure 4. An example for the composition, based on ontologies, of a query for training in risk prevention

- 2.1 Check if the expression contains only Boolean or grouping operators (AND, OR, NOT, (,)) or "=" operator (the last one, for concept instance assignment);
- 2.2 Check if the instance assigned to a concept C is defined in the same ontology the concept C belongs to and is the concept instance (because it is possible that an instance of another concept was dragged as value, by mistake);
- 2.3 For each concept in condition, check for the concept (direct or indirect) relationship with at least another concept in the condition. This check is needed because, for concepts in condition that are not correlated (e.g., if in the model there is no relationship between "Laboratory Procedure" and "Substance with micro-organisms"), after the search execution, the resulted concepts might not be correlated or are redundant (e.g., the discovered risks are for all laboratory procedures, not only for those that use substances with micro-organisms).
3. *Analyze the expression for the Descriptive Restrictions* for its syntactic and semantic correctness:
 - 3.1 Check if the expression contains only attributes of concepts in ontologies or attributes of relationships, possibly assigned with constant values;
 - 3.2 Check if in the expression, the relational, arithmetic, string, Boolean or grouping operators are correctly used and combined;
 - 3.3 Check on the existence of a relational expression for any attribute (it should be followed by a relational operator and by a value (constant expression));
 - 3.4 Check if the value assigned to an attribute complies with the data type defined in ontology/ model for the respective attribute.
4. *Semantic completion of Search Query:*
 - 4.1 For each *ontology* qO given by its name in the *Search Query*,
 - Search for all *concepts* in qO connected (directly or indirectly) with at least a concept in the *Associative Condition*; Store them in a temporary repository.

- 4.2 For each *concept* qC given by its name in Search Query,
- Search for all *subtypes* (only in specialization ontologies) and *synonyms* of it, connected (directly or indirectly) with concepts in the Associative Condition; Store them in a temporary repository.
5. **Filter the concepts** resulted from Step 4 (and stored in *the* temporary repository), according to:
- the attribute restriction in Descriptive Restrictions expression;
 - the rules on the selected model.
 - the rules on each ontology named in the Search Query (by itself or by its concepts).

The system deletes the concepts from the temporary repository if they do not comply with the restrictions and rules.

6. **Search and filter the instances** of the concepts resulted from Step 5, by the relationship with instances in the Associative Condition and in rules.
7. **Compose the result.** For each *concept* in the temporary repository resulted from Step 4 and 5, order the resulted concepts and instances, according to Search query:
- For ontology name: order the concepts in ontology and their instances;
 - For an ontology concept: order its instances.

4. Conclusion

The paper describes the semantic framework of a system for risk prevention. It relies on a dedicated reference ontology and model, on domain specific ontologies and on reasoning on them, basically, for the search and discovery of prevention risks, documents and actions previously registered in the system.

Although the importance of the ontologies and of a model for risk prevention has already been revealed in the literature, there is no general software for on-line training, the goal of the system presented in this paper.

The system architecture was adapted to a semantics-based view on the risk prevention. Its *interface dedicated to the domain experts* moves the work for ontology editing and risk design, from IT experts to the domain experts. The system and its portal will

contribute to a *knowledge repository for risk prevention* inside and cross organizations. It will be accessible from Web and will gradually replace the periodical training in organizations.

The risk prevention model described in the paper can be dynamically extended with new ontologies, relationships, constraints and rules, when necessary. They will be automatically considered in future inferences on the model.

The system implementation status is as follows. In the *platform for the risk design*, there are implemented the *Ontology editor* (for specialization, composition and list-like ontologies), with automatic inheritance of the attributes only for the specialization ontologies; and the (*reference and domain*) *Model editor*. These tools provide the graphical view of the models and ontologies and help the designers:

- add to the reference model: new ontologies, new inter-ontology relationships, new attributes for ontologies and relationships.
- add to the domain model and ontologies: new concepts, new concept instances, new relationship instances.

Rule editor and *Query template editor* on this platform are under development.

The platform for risk evaluation and decision-making is partly implemented: *Ontology navigator* is finished; *Model navigator*, *Query editor*, *Inference engine* and *Query result generator* are under development.

REFERENCES

1. COPPÉE, G., **Occupational Health Services and Practice**, in J. Mager Stellman (Ed.), *Encyclopaedia of Occupational Health and Safety*, Geneva, PA: International Labour Office, 1998.
2. METHODWARE, **ISO 31000: Risk Management Standard**, 2009, www.methodware.com/iso-31000-risk-management-standard-published [visited, May 14, 2010].
3. ISO, **ISO 31000: 2009 Risk management -principles and implementation of risk management**,

- www.iso.org/iso/catalogue_detail.htm?csnumber=43170 [visited, May 10, 2010].
4. ISO, **ISO/ IEC Guide 73:2002 “Risk management — Vocabulary Guidelines for use in Standards**, 2009, www.iso.org/iso/catalogue_detail?csnumber=34998 [visited, May 2010].
 5. MAHB (Major Accident Hazards Bureau), **Safety Management Systems - Seveso II**, Official Publications of European Communities, Luxembourg, 1998, <http://mahbsrv.jrc.it/GuidanceDocs-SafetyManagementSystems.html> [visited, May, 2010].
 6. LEE, S.-W., R. GANDHI, D. MUTHURAJAN, D. YAVAGAL, G.-J. AHN, **Building Problem Domain Ontology from Security Requirements in Regulatory Documents**, Proc. Intl. WS on Software Engineering for Secure Systems. ACM Press, 2006.
 7. TSOUMAS, B., D. GRITZALISI, **Towards an Ontology-based Security Management**, Proc. Intl. Conf. on Advanced Information Networking and Applications (AINA), Vienna, 2006.
 8. KLEMEN, M., E. WEIPPL, A. EKELHART, S. FENZ, **Security Ontology: Simulating Threats to Corporate Assets**, Proc. 2nd Intl. Conf. on Information Systems Security (ICISS), Springer, 2006.
 9. SIMMONDS, A., P. SANDILANDS, L. VAN EKERT, **An Ontology for Network Security Attacks**, Proc. Asian Applied Computing Conference (AACC), LNCS, vol. 3285, Springer, 2004.
 10. MITRA P., C. PAN, P. LIU, V. ATLURI, **Privacy-preserving Semantic Interoperation and Access Control of Heterogeneous Databases**, Proc. Symposium on Information, Computer and Communications Security, ACM Press, 2006.
 11. FORCADA, N., M. CASALS, A. FUERTES, **The Basis of a Decision Making Tool for Risks’ Evaluation Based on Ontologies**, Proc. Intl. Conf. on Inf. and Knowledge Management - Helping the Practitioner in Planning and Building, Stuttgart, 2007, www.baufachinformation.de/aufsatz.jsp?ul=2008031001259 [visited, April 2010].
 12. KOLA, J., B. WHEELDIN, A. RECTOR, **Lessons in Building OWL Ontology driven applications: OCHWIZ – an Occupational Health Application**, National e-Science Centre, 2007, www.allhands.org.uk/2007/programme/download490f.html?id=849&p=paper [visited, April 12, 2010].
 13. W3C, **OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax**, Oct. 2009, www.w3.org/TR/2009/REC-owl2-syntax-20091027/ [May, 2010].
 14. SWUSTE, P., **Qualitative Methods for Occupational Risk Prevention Strategies in Safety or Control Banding safety**, Safety Science Monitor, Issue 3, vol. 11, 2007.
 15. GANAYE, P., **An Expression Evaluator written in VB.NET**, www.codeproject.com/KB/vb/expression_evaluator.aspx
 16. ABILOV, V., **WYSIWYG Rule Editor: Create and Test Rules for any .NET Type**, <http://bloggingabout.net/blogs/vagif/archive/2009/04/13/wysiwyg-rule-editor-create-and-test-rules-for-any-net-type.aspx>
 17. ALEXANDRU, A., F. FILIP, A. GALATESCU, E. JITARU, **Using Ontologies in eHealth and Biomedicine**, in book A. Shukla and R. Tiwari (Eds) "Intelligent Medical Technologies and Biomedical Engineering: Tools and Applications", IGP Global, May, 2010.
 18. GALATESCU, A., A. ALEXANDRU, C. ZAHARIA, S. KOVACS, **Ontology-based Modeling and Inference for Risk Prevention**, Proc. Intl. Conf. on Advances in Semantic Processing, SEMAPRO, Florence, Italy, October, 2010
 19. ABOUT. **about Suite for ASP.Net** www.about.com, [visited, November 17, 2010]