Process Systems Engineering Laboratory, Tokyo Institute of Technology, Technical Report TR-2003-01, April 2003

An Ontological Approach to Represent HAZOP Information

Kiyoshi Kuraoka and Rafael Batres

Content Areas: hazard analysis, process engineering ontology, SUMO, semantic web,

Abstract

To overcome the limitations of text-based descriptions a HAZOP ontology has been proposed that provides a basic set of standard concepts and terms The development of the ontology uses the Upper level Ontology, SUMO (The Suggested Upper Merged Ontology) and a Process Engineering Ontology, that define general-purpose terms and act as a foundation for more specific domains. The ontology is developed so that engineers can build new concepts out from the basic set of concepts. This paper evaluates the proposed ontology by means of use cases that measure the performance in finding relevant information used and produced during the safety analyses. In this paper, the extraction of knowledge is performed using JTP (An object oriented Modular Reasoning System) that is used for querying the ontology.

1 Introduction

Safety plays a very important role throughout the life cycle of a chemical plant. To ensure safety and minimize later plant changes, risk analyses are performed during process and plant design stages. HAZOP (Hazard and Operability Analysis) is probably one of the most widely used methods of safety evaluation. However information that is used and produced during the HAZOP studies are recorded in the form of text-based documents. Consequently, the reuse of this knowledge during design or during operations is limited as a result of the difficulties in finding, retrieving, and analyzing HAZOP-related information.

A number of tools are available in the market to support the documentation of the HAZOP sessions. However, the information stores in these tools is in the form of textual natural language descriptions that limit the computer-based extraction of knowledge for the reuse of the HAZOP analyses in other designs or during plant operation. To overcome the limitations of text-based descriptions, a HAZOP ontology has been proposed that provides a basic set of standard concepts and terms The development of the ontology uses the Upper level Ontology, SUMO (The Suggested Upper Merged Ontology) and a Process Engineering Ontology, that define general-purpose terms and act as a foundation for more specific domains. The ontology is developed so that engineers can build new concepts out from the basic set of concepts. This paper evaluates the proposed ontology by means of use cases that measure the performance in finding relevant information used and produced during the safety analyses. In particular, the extraction of knowledge is performed using JTP (An object oriented Modular Reasoning System) that is used for querying the ontology. While a few sample queries are included to show the use of the ontology several more are described in [Kuraoka, 2003].

2 HAZOP

HAZOP (Hazard and Operability Analysis) is one of the most widely used safety analysis techniques widely used in the process industries. Having its origins in ICI in the 1960s, HAZOP seeks to find underlying hazards associated to the process and plant and then identifies causes and consequences of possible deviations from the design intention [Kletz, 1999]. Typically, in HAZOP analyses a team of specialists examines the P&IDs of the plant and applying a series of guidewords to each pipeline (the connected pipes and other plant devices that join two main plant items). Important features of this technique are: description of the intended characteristics of the process and plant, deviations from the intent, causes of deviations, consequences (process drifts, equipment malfunctions, failures and operating difficulties).

3 Ontologies

Ontologies describe a shared and common understanding of a domain that can be communicated between people and heterogeneous software tools. Moreover, ontologies constitute the basis of a new generation of the World Wide Web known as Semantic Web, where software agents and people can share and exchange data in a way that all the involved parties share the same meaning of the terms describing the data [Berners-Lee, et al. 2001]. From the point of view of information modeling, ontologies make a commitment to an unambiguous representation of the concepts of a specific domain of discourse rather than to the structure of a data container. The objectives of developing an ontology are:

- To facilitate sharing/exchange of information and knowledge
- To support integration of tools
- To provide the same perspectives with collaborating teams and tools,
- To create a common vocabulary,
- To describe unambiguous definitions that both computers and teams can understand.

An ontology is constructed by defining classes, their taxonomy, relations or properties, and axioms.

A number of ontology languages have been developed with a variety of expressivity and robustness, including KIF [Genesereth and Fikes, 1992], Ontolingua [Farquhar, Fikes, and Rice, 1997], and DAML+OIL [McGuinness *et al.*, 2002]. In this paper, DAML+OIL has been selected based on its adequacy to Internet-based communications, the number of free editing tools, and the efficiency in performing inferences with today's inference engines.

DAML+OIL has its roots in description logic and has been proposed as starting point for the W3C Semantic Web Activity Ontology Web Language (OWL) [Smith *et al.* 2003].

5 Upper ontologies

Upper ontologies define top level concepts such as Processes, Objects, Mereological and Topological concepts from which more specific classes and relations are defined, including Physicochemical Processes, Substances (such as Material, Material Flows), Devices (such as Equipment Item, Equipment Connection), Organizational Tasks, Plant Operation. An earlier version of a process engineering ontology was specified in Ontolingua (classes and relations) and Knowledge Interchange Form (axiom definitions) had strong dependencies on the ontology development environment which reduced the possibilities of a wide-spread use of the ontology [Batres and Naka, 2000].

OMPEK, a revised version of the process engineering ontology is being developed that is based on the Suggested Upper Merged Ontology (SUMO) which is an upper ontology that is been developed by a diverse group of collaborators from the fields of engineering, philosophy, and information science [Niles and Pease, 2001]. DAML+OIL has been selected as a format to encode the ontologies mostly based on its computational efficiency. Consequently, the original SUMO ontology that is originally encoded in KIF has been translated to DAML+OIL. In the rest of the paper, namespaces of the form ONTOLOGY:CONCEPT are used to identify the origin of a certain concept or relation.

5.1 SUMO

The Suggested Upper Merged Ontology (SUMO) is an upper ontology that is been developed by a diverse group of collaborators from the fields of engineering, philosophy, and information science.

The SUMO ontology was created by merging and reorganizing publicly available ontologies such as Russell and Norvig's uppper ontology [Russell and Norvig, 1995], John Sowa's upper level ontology [Sowa, 2000], the ontologies available at the Stanford Ontology Editor, and several mereotopogical theories, including Peter Simons' mereological theory. The ontology is encoded in a simplified version of KIF (Knowledge Interchange Format) known as SUO-KIF. The development of SUMO was intended to be applicable in engineering-oriented contexts.

Entity is the root concept in SUMO that encompasses Physical and Abstract classes. Instances that belong to Physical are entities that have a location in space-time. Instances of Abstract can be said to exist in the same sense as mathematical entities such as sets and equations, but they cannot exist at a particular place and time without some physical encoding or embodiment.

Physical entities are divided into *Object* and *Process* classes. Object entities are defined as things that are present at any moment of their existence. Examples include normal physical objects, geographical regions. On the other hand, process entities are the class of things that happen and have temporal parts or stages. Examples include engineering activities, and chemical reactions.

Abstract entities are divided into *Set*, *Proposition*, *Quantity*, and *Attribute*.

5.2 OMPEK (Ontologies for Modeling Process Engineering Knowledge)

The OMPEK ontology is being developed as a free, public standard ontology for the process-engineering domain.

OMPEK aims to cover areas such as substances, material processes, production plans and operation, processing equipment, human systems and value-chain components. The objective is to propose ontologies that allow extensions for applications in specific areas of process engineering. For example, the standard ontology will be able to define the concept of PlantDevice to describe equipment and plant devices but specific classes such as RotatoryPump will not be part of the core ontology although they can be defined in optional libraries or in other projects that reuse OMPEK.

OMPEK is being developed in DAML+OIL and it will be available as Open Content to facilitate the release of enhanced versions of the ontology in freely available, high-quality, well-maintained fashion.

The main theories in OMPEK are quantities, physicochemical processes, plant devices, substances and intentional processes.

The ontology separates presentation from representation, which translates into two views of a physical entity (such as a plant device), namely a view that represented the actual object and a view that represented an abstract description of the entity. Ports and connection ports only exist in the latter view. This is consistent with the ontological definitions in SUMO. The actual object is represented by SUMO:EngineeringComponent and ports can be described using the graph concepts defined in SUMO. A number of mereotopological concepts are already defined in SUMO for describing connectivity and part-of relationships.

4 HAZOP Ontology

The methodology used in the development of the ontology was as follows. Firstly, several variations of the HAZOP procedure were identified and reconciled with the assistance of experts from two engineering companies and one process development organization. Typical use cases where identified that identified key information items which were then considered as candidates for the concepts in the ontology. A case study worked out by one of the safety experts was taken as an example for representing instances used in querying the ontology for testing purposes. The overall structure of the HAZOP ontology is shown in Figure 1.

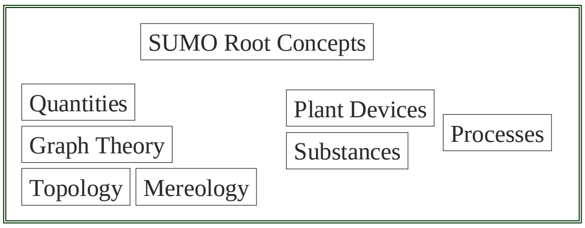

Figure 1. Main theories used in the Ontology

4.1 Quantities

The basic concepts about quantities are defined in SUMO. *Quantity* under *Abstract* subsumes the concepts of *Number* and *PhysicalQuantity* . *Number* is a *Quantity* independent of a measurement system while *PhysicalQuantity* is a *Quantity* consisting of a *Number* and a given unit of measure (*UnitOfMeasure*).

Following the arguments presented in … we support the idea that *Objects* and *Processes* should not be allowed to define quantities as attributes as quantities are not an inherent property of a *Physical*. Consequently, OMPEK introduces the concept of *QuantityFunction* that maps one or more instances of *Physical* (and possibly one or more instances of PhysicalQuantity) to a *PhysicalQuantity*. The concept of *QuantityFunction* is useful when describing several measurements of the same quantity that are for instance taken by different instruments. In addition, *QuantityFunction* can represent quantities of an object that change in time. This is done by means of the functionalDomainQuantity property as shown in Figure 2.

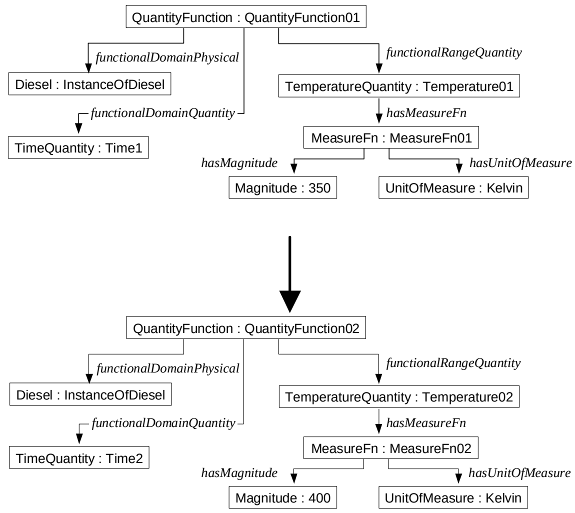

Figure 2. Example of the use of QuantityFunction.

4.2 Processes

SUMO defines concepts and relations such as *subProcess*, *patient*, *instrument*, *resource*, *causes*, *result*, *QuantityChange*, *Increasing*, *Decreasing*. The SUMO:*patient* of a process is the entity that plays the role of participant in the process that may be moved, modified, etc. The SUMO:*instrument* of a process refers to the tool that is used by an agent in bringing about event and that tool is not changed by event. For example, the actuator of a valve is an instrument of opening that valve. The SUMO:*resource* of a process means that resource is present at the beginning of process, is used by process, and as a consequence is changed by process.

The causation relation between instances of SUMO:*Process*. (*causes* process1 process2) means that the instance of Process process1 brings about the instance of Process process2, e.g. (*causes* Cavitation PumpFailure). Similarly, the SUMO:*result* of a process means that an object is a product of process. The outcome of a process is defined by means of the *result* relation. For example, (result PlantConstruction MyPlant) indicates that the instance MyPlant is the result of the instance PlantConstruction. The origin of a process indicates the source where the process began.

SUMO:*Increasing* is any QuantityChange (a kind of Process) where the PhysicalQuantity (patient of increasing)is increased. Decreasing refers to any QuantityChange (a kind of Process) where the PhysicalQuantity (patient of cecreasing)is decreased.

In order to reason about physical and chemical phenomena OMPEK extends SUMO by incorporating the concepts *PhysicochemicalProcess* and *takesPlaceIn*. *PhysicochemicalProcess* refers to the physical or chemical phenomena that are manifested through changes in the attributes of a substance. One or more instances of PhysicochemicalProcess take place in instances of Object. For example, a given chemical reaction takes place in a certain reactor.

4.3 Mereo-Topology

Mereo-topological concepts are defined in SUMO which are mainly based on the Classical Extensional Mereology as described in [Simons, 1987].

Mereology expresses the part-whole relations of an *Object*, which means that a component can be decomposed into parts or subcomponents that in turn can be decomposed into other components. Topology refers to the connectivity between *Objects*. All the mereological relations in SUMO are derived from the *part*. (*part* part whole) means that the *Object* part is part of the *Object* whole. The relation *part* also implies that very Object is a part of itself. The subclass of *part* engineeringSubcomponent means that an EngineeringComponent is structurally a properPart of another EngineeringComponent in which the two *EngineeringComponents* cannot be subcomponents of each other.

Connection is specified by means of the use of the SUMO terms *connected*, *connects*, *connectedEngineeringComponents*, *connectsEngineeringComponents*. The binary relation connected is the minimal and most general relation between connected components. The concept of *connectedEngineeringComponents* is a subclass of *Connected* that constrains the connected objects by not being able to be an *engineeringSubComponent* of the other.

4.4 Plant Devices

The physical part of the plant (the hardware) is defined by means of instances of *Plant* and *PlantDevice*. *PlantDevice* which is a subclass of SUMO:*Device* is used to describe a equipment items, fittings and mechanical parts but also manufacturing plants, or processing complexes. Equipment which is a subclass of *PlantDevice* and a subclass of SUMO:*EngineeringComponent* is specific to equipment items such as compressors, pumps, etc.

4.5 Graph Theory

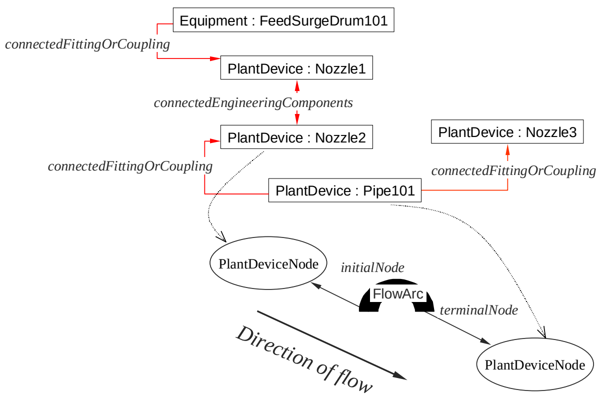

Figure 3. Graph representation of devices

A graph representation is used for mereological and topological reasoning about composite objects. A graph is composed of GraphNodes and GraphArcs. Directed graphs where every arc has direction are represented by using the relations *initialNode* and *terminalNode*.
A PlantDeviceNode is a graph representation of a *PlantDevice*. OMPEK:*FlowArc* indicates the intended direction of material flow along two pieces of equipment represented as instances of *PlantDeviceNode*.

4.6 Substances

Substance in SUMO is defined as a SUMO:*Object* that has only arbitrary pieces as parts and any parts have some properties that are similar to those of the whole. A substance always coexists with an instance of SUMO:*PhysicalState* that is reified into *Solid*, *Liquid* and *Gas*. Subclasses of SUMO:*Substance* include SUMO:*Mixture* and SUMO:*PureSubstance*.

The relation OMPEK:*chemicalComponent* which is a subclass of SUMO:*piece* is used in for reasoning about the components of a *Mixture*.

Substances that participate in OMPEK:*PhysicochemicalProcesses* can be characterized using the relation OMPEK:substanceInProcess which is a subclass of SUMO:*patient*. Mappings between OMPEK:*PlantDevice* and SUMO:*Substance* are possible by means of reifications of the relations SUMO:*located* and its subclass SUMO:*contains* (inverse: OMPEK:*contained*). *Contains* is disjoint to *part*.

4.7 HAZOP deviations

People working in HAZOP analysis formulate deviations by combining guidewords such as none, more, less and quantities (typically -but not restricted to- those known as process variables). From an ontological point of view, deviations can be modeled as processes, specifically as SUMO:QuantityChanges. Deviations using the guide word *more* can be modeled as SUMO:*Increasing* (a SUMO:*QuantityChange* where the *PhysicalQuantity* is increased.). Similarly, deviations formulated with the guide word less can be modeled as SUMO:*Decreasing* (a SUMO:*QuantityChange* where the *PhysicalQuantity* is decreased).

Specific deviations such as *more than* (more components present in a mixture, more phases present, etc.) are formulated with similar processes, namely OMPEK:*NumericalIncreasing* and OMPEK:*NumericalDecreasing*.

4.8 Abnormal Processes

Abnormal situations result when there is at least a QuantityChange of the *PhysicalQuantiy(ies)* of a PhysicochemicalProcess that increases the likelihood of the creation of processes such as SUMO:*Damaging* of Equipment, SUMO:*Building*s, or People. In other words, an AbnormalProcess characterized by at least a QuantityChange of PhysicalQuantity SUMO:results into a *Damaging* process. Damaging is the class of processes where the patient no longer functions normally or as intended. Informally, a *QuantityChange results* in *Damaging* and *Damaging* may result in another *QuantityChange*.

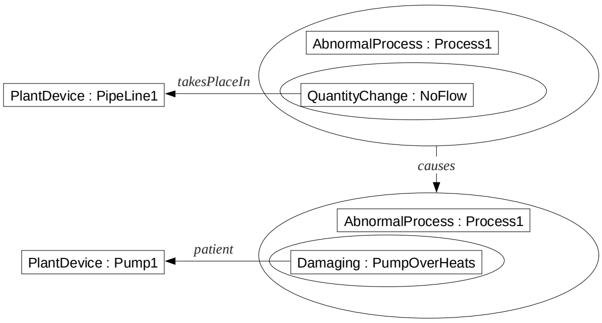

Figure 4. Example of information instances describing abnormal processes

4.9 Causes and Consequences

Causes and consequences can be modeled using the SUMO:*causes* and SUMO:*results*.

With the representation of abnormal processes described in the previous subsection, contrary to current text—based approaches in HAZOP, an information system can verify that information should be supplied about requires information that specifies whether the patient of consequence is a *PlantDevice*, a *Substance*, *Person*, etc.

4.10 Operations, Maintenance, Corrective Actions

Operations, maintenance activities and actions that prevent or correct an abnormal situation are defined as SUMO:*Process*(es). Specifically, OMPEK defines *Operation* as an SUMO:*IntentionalProcess* that causes a *QuantityChange* in a *Quantity* associated to a part of a *PlantDevice*. A SUMO:*IntentionalProcess* is a subclass of Process that has a specific purpose. Operating procedures, batch recipes, maintenance procedures, production plans or schedules are modeled as subclasses of SUMO:*Plan*. A *Plan* is a specification of a sequence of *Processes* which is intended to satisfy a specified purpose at some future time.

5 Queries to the ontology

Queries to the ontology are formulated using JTP (Java Theorem Prover) which is a reasoning system that can interpret DAML+OIL files. JTP translates each DAML+OIL statement into a KIF sentence of the form (PropertyValue Value Predicate Subject Object). Then it simplifies those KIF sentences using a series of axioms that define DAML+OIL semantics. DAML+OIL statements are finally converted to the form (Predicate Subject Object). Queries are formulated in a format similar to KIF, where variables are preceded by a question mark. Following are some of the queries to the knowledge base. Namespaces have been omitted in the sake of clarity and space.

What are the hazards produced in case there is a fault in the pump FeedPump?

```
> ask
Enter a query:

(and
   (takesPlaceIn ?FAULT FeedPump)
   (causes ?FAULT ?FAILURE)
   (subProcess ?FAILURE ?ABPROCESS1)
   (causes ?ABPROCESS1 ?ABPROCESS2)
   (subProcess ?DEVIATION ?ABPROCESS2)
   (takesPlaceIn ?DEVIATION ?LOCATION)
   (causes ?DEVIATION ?CONSEQUENCE))

Query succeeded.

Bindings 1:
?FAULT = |Mortorfailure|
?FAILURE = |MotorStops|
?DEVIATION=|HighLevelOfD-201|,
?LOCATION= |FeedSurgeDrum|,
?CONSEQUENCE = |Overflow|

Bindings 2:
?FAULT = |Cavitation-1|
?FAILURE = |PumpDamage|
?DEVIATION = |HighLevelOfD-201|
?CONSEQUENCE = |Overflow|
?LOCATION = |FeedSurgeDrum|

…
```

What are the primary and secondary effects of the consequences of an abnormal situation in the FeedSurgeDrum?

```
> ask
Enter a query:
(and
   (takesPlaceIn ?DEVIATION FeedSurgeDrum)
   (causes ?DEVIATION ?CONSEQUENCE)
   (subProcess ?DEVIATION ?ABPROCESS1)
   (causes ?ABPROCESS1 ?ABPROCESS2)
   (subProcess ?SECONDARYEFFECT ?ABPROCESS2)
   (takesPlaceIn ?SECONDARYEFFECT ?LOCATION)
   (causes ?SECONDARYEFFECT
            ?TERTIARYEFFECT))

Query succeeded.

Bindings 1:
?DEVIATION =|HighLevelOfD-201|
?CONSEQUENCE =|TankOverflow201|
```

?SECONDARYEFFECT = |FlareSystemUpset01|
?TERTIARYEFFECT=|FlamingLiquidOutsideTheFlareStack01|
?LOCATION =|FlareLine01|

What is the PhysicalState of the mixture that is involved in the |FlareSystemUpset01|? What are the substances of the mixture?

```
> ask
Enter a query:
(and (materialInProcess FlareSystemUpset01 ?SUBST)
(chemicalComponent ?SUBST ?COMPONENT)
(physicalState ?SUBST ?PHASE))

Query succeeded.
Bindings 1:
?COMPONENT =|RawDiesel|、?PHASE = |Liquid|
```

6 Conclusions

A working version of a process engineering ontology written in DAML+OIL has been used in an effort to improve the representation of knowledge that is used and produced during Hazards and Operability Studies (HAZOP). Traditionally, the information that is used and produced during the HAZOP studies is registered in text format. The reusability of this knowledge during design or operations is limited due to difficulties in finding and analyzing information. The basic ontology has been extended so that engineers can use more informative queries (instead of text based) to find relevant information during the safety analyses.

Very expressive languages such as SUO-KIF (the language used in SUMO) provide flexible ways to express statements involving complex concepts such as purposes, and time-related concepts. On the other hand, DAML+OIL sacrifices expressivity for efficiency which nevertheless is an important requirement in an industrially deployed ontology. Future work will identify the extent to which complex concepts relevant to process engineering can be described in DAML+OIL terms.

References

[Batres and Naka] Rafael Batres and Yuji Naka. Process Plant Ontologies based on a Multi-Dimensional Framework. In Foundations of Computer Aided Process Design, AIChE Symposium Series, No. 323, ISBN No: 0-8169-0826-5 (2000)

[Berners-Lee, et al. 2001] Tim Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. Scientific American, May, 2001.

[Genesereth and Fikes, 1992] Genesereth, M. R., Fikes, R. E. (1992). KIF Version 3.0 Reference Manual, [Online] Available: http://logic.stanford.edu/papers/kif.ps

[Farquhar, Fikes, and Rice, 1997] Farquhar, A., R. Fikes, J. Rice. Tools for Assembling Modular Ontologies in Ontolingua. Technical Report KSL-97-03, Stanford University, KSL

[Kletz, 1999] Trevor Kletz. Hazop and Hazan: Identifying and Assessing Process Industry Hazards. Institution of Chemical Engineers, 1999.

[Kuraoka, 2003] Kiyoshi Kuraoka: An Ontological Approach to Represent HAZOP Information. Tokyo Institute of Technology, Process Systems Engineering Laboratory, Masters Thesis, March 2003

[McGuinness *et al.*, 2002] Deborah L. McGuinness, Richard Fikes, James Hendler, Lynn Andrea Stein. DAML+OIL: An Ontology Language for the Semantic Web. IEEE Intelligent Systems, Vol. 17, No. 5, pp. 72-80, 2000.

[Niles and Pease, 2001] Ian Niles and Adam Pease. Towards a Standard Upper Ontology. *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems* (FOIS-2001), Ogunquit, Maine, October 17-19, 2001.

[Russell and Norvig, 1995] Stuart J. Russell and Peter Norvig. Artificial Intelligence: A Modern Approach. Prentice-Hall, 1995.

[Simons, 2000] Peter Simons. Parts: A Study in Ontology. Oxford University Press, 2000

[Smith *et al.* 2003] Michael K. Smith, Chris Welty, Deborah McGuinness. Web Ontology Language (OWL) Guide Version 1.0, W3C Working Draft 10, February 2003. [Online] http://www.w3.org/TR/owl-guide/

[Sowa, 2000] John Sowa. Knowledge Representation: logical, philosophical, and computational foundations. Brooks/Cole, 2000.