An Ontology Approach to Support FMEA Studies

V. Ebrahimipour, PhD, University of Tehran

K. Rezaie, PhD, University of Tehran

S. Shokravi, University of Tehran

Key Words: FMEA, ontology based approach, owl, protégé, rdf, xml

SUMMARY & CONCLUSIONS

FMEA (Failure Modes and Effects Analysis) is a method to analyze potential reliability problems in the development cycle of the project, making it easier to take actions to overcome such issues, thus enhancing the reliability through design. FMEA is used to identify actions to mitigate the analyzed potential failure modes and their effect on the operations. Anticipating these failure modes, being the central step in the analysis, needs to be carried on extensively, in order to prepare a list of maximum potential failure modes. However, the information stored in risk assessment tools is in the form of textual natural language descriptions that limit computer-based extraction of knowledge for the reuse of the FMEA analyses in other designs or during plant operation. To overcome the limitations of text-based descriptions, FMEA ontology has been proposed that provides a basic set of standard concepts and terms. The development of the ontology uses an upper ontology based on ISO-15926, which defines general-purpose terms and act as a foundation for more specific domains. The ontology is developed so that engineers can build new concepts 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.

1 INTRODUCTION

Risk management is a central part of any organization's strategic management. It is the process whereby organizations methodically address the risks attached to their activities with the goal of achieving sustained benefit within each activity and across the portfolio of all activities. The focus of good risk management is the identification and treatment of these risks. Its objective is to add maximum sustainable value to all the activities of the organization. It marshals the understanding of the potential upside and downside of all those factors which can affect the organization. It increases the probability of success, and reduces both the probability of failure and the uncertainty of achieving the organization's overall objectives.

Failure Mode and Effect Analysis (FMEA) is yet another powerful tool used by system safety and reliability engineers/analysts to identify critical parts, functions and components whose failure will lead to undesirable outcomes such as production loss, injury or even an accident. The tool was first proposed by NASA in year 1963 for their obvious reliability requirements. Since then, it has been extensively used as a powerful technique for system safety and reliability analysis of products and processes in wide range of industries—particularly aerospace, nuclear, automotive and medical. The main objective of FMEA is to discover and prioritize the potential failure modes (by computing respective RPN), which pose a detrimental effect on the system and its performance. The results of the analysis help managers and engineers to identify the failure modes, their causes and correct them during the stages of design and production.

The process for conducting a Failure Modes and Effects Analysis is summarized as follows:

1. Describe product or process
2. Define Functions
3. Identify Potential Failure Modes
4. Describe Effects of Failures
5. Determine Causes
6. Direction Methods or Current Controls
7. Calculate Risks
8. Take Action
9. Assess Results

It should be noted that a failure mode may be introduced after any change and updates are made to the product and process. Thus, FMEA might need to be reviewed (and updated) whenever a new product (or process) is being introduced, any changes are made to the operations or a change is made to the process design. By providing the engineers with a tool to assist in ensuring reliable and safe products and processes, FMEA grants certain benefits for project management. It emphasizes problem prevention and acts as a catalyst for teamwork and exchange of healthy ideas. It captures engineering knowledge and provides a focus for improved testing and development, eventually resulting in increased customer satisfaction.

2 ONTOLOGIES

Ontology describes 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

1-4244-2509-9/09/$20.00 ©2009 IEEE

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 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.

Ontology is constructed by defining classes, their taxonomy, relations or properties, and axioms. A class represents a category of similar things that share a set of properties. A relation is a function that maps its arguments to a Boolean value of true or false. Examples of relations are less-than, greater-than, and part-of. Class taxonomies are defined with the use of the subclass relation. A class is a subclass of another class if the former represents a set of things that subsumes the set of things represented by the latter.

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), DAML+OIL (McGuinness et al., 2002) and more recently OWL (W3C, 2004). In this paper, OWL (Web Ontology Language) 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.

3 UPPER ONTOLOGY

Upper ontologies define top-level concepts such as physical objects, activities, mereological and topological relations from which more specific classes and relations can be defined. Examples of upper ontologies include SUMO (Niles and Pease, 2001), Sowa upper ontology (Sowa, 2000), Dolce (Gangemi et al., 2000) and CliP (Bayer, 2003). Engineers can start by identifying key concepts by means of activity modeling, use cases and competency questions. These concepts are then defined based on the more general concepts provided by the upper ontology. This avoids reinventing the wheel with better integration and maintenance.

As an effort to support the development of domain ontologies, core concepts used in the ontology are based on the ISO-15926 standard. In its original form, ISO-15926 defines a common data model for long-term data integration, access and exchange. It was developed in ISO TC184/SC4-Industrial Data by the EPISTLE consortium (1993-2003) and designed to support the evolution of data along time.

3.1 Top level concept

"thing" is the root concept in ISO-15926 that subsumes abstract_object and possible_individual classes. A thing is anything that is or may be thought about or perceived, including material and non-material objects, ideas, and activities. Every thing is either a possible_individual, or an abstract object (see Figure 1). Members of possible individual are entities that exist in space and time, including physical objects like a bowl or ideas that exist in our imagination. Individuals that belong to abstract_object can be said to exist in the same sense as mathematical entities such as numbers or sets but they cannot exist at a particular place and time.

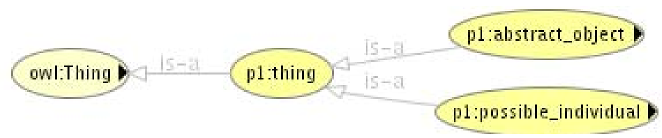

Figure 1 - Top-level concepts

possible_individual is divided into arranged_individual, actual_individual, whole_life_individual, activity, physical_object, period_in_time and event (see Figure 2).

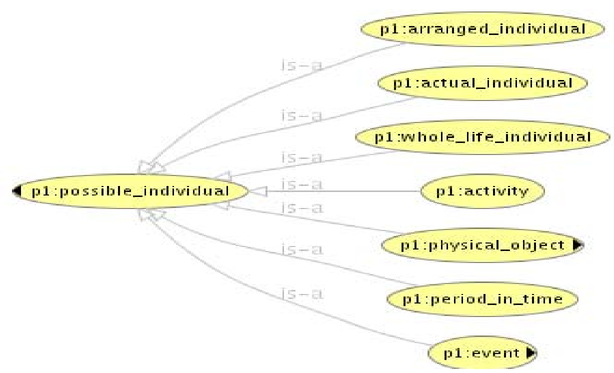

Figure 2 - Subclasses of possible_individual

3.2 Activities and causality relations

Activities can have temporal boundings linking events as well as points_in_time because activity is a subclass of possible_individual. For example, production steps in a manufacturing schedule have a start time and end time. Activities bring about change by causing an event. Events mark the beginning, or the ending of a possible_individual. An activity consists of the temporal parts of those members of possible_individual that participate in the activity. For example, a meeting activity shares the temporal parts of the persons that gather together to exchange ideas.

3.3 Physical objects

A physical_object is a possible_individual that is a distribution of matter, energy, or both. Examples of physical_object are a table, a pump, a piece of metal, a laser beam (Figure 3).

Physical objects can be instances of arranged_individual which defines those possible individuals that have parts each

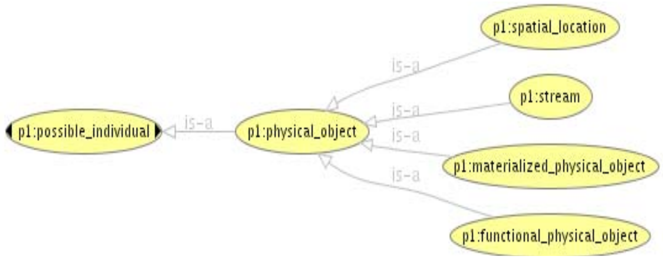

Figure 3 - Physical object classes

of which playing a distinct role. Instances of physical objects can be related to instances of abstract. For example, the water contained in my glass has the phase liquid. In the ontology, phase is abstract.

4 PROTÉGÉ

To present the FMEA and the ontology example in this paper, protégé software was used, which was developed by Stanford University. In this example the software is used to harvest RDF/XML codes.

Protégé is one of the ontology editors with import and export capability and knowledge acquisition and ontology development tool which developed by SMI, Stanford University. It contains the following ontology components:

- Classes represent entities of the domain. They can be concrete or abstract, that is, they can have direct instances or not respectively. Slot constraints can be attached to the class, and determine constraints on the slot values of the class instances.
- The taxonomic relation subclass-of can be defined between classes, allowing multiple classifications. The relation subslot-of can be defined between slots.
- Slots represent interactions between domain objects or characteristics of class instances. They have value types, minimum and maximum cardinalities, minimum and maximum values in case of numeric slots, default values, template values, constraints and the inverse slot. Slots are defined independently of the classes to which they are attached, and can be attached in two different ways:
  - ➢ Template slots. They describe properties of the instances of the class to which they are attached or interactions between instances of the class and instances of other classes. They are inherited by the subclasses and instances of the class, where they can be constrained and/or take values.
  - ➢ Own slots. They describe properties of the class itself or interactions between the class and other classes, taking some values in the class. These values are not inherited by its subclasses nor by its instances. The template slots of a class become own slots in its instances.
- Facets define slot constraints. The features of slots defined before (value types, minimum and maximum cardinalities, etc.) are built-in facets in the Protégé model. However, new facets can be defined for them.
- Instances of classes define individuals in the domain. They contain own slots (which are the template slots of the classes from which they are instances) with their corresponding values. In Protégéinstances belong only to one class. Constraints are first order logic sentences used to check constraints in the ontology. They are created with PAL (Protégé Axiom Language), which is a superset of first order logic.

One of the main features of the Protégé is that it allows defining metaclasses, classes that act as templates for creating other classes. In fact, metaclasses are used to define the own slots of classes in an ontology. Own slots are defined as template slots in the metaclass, and later the classes that are instances of the metaclass inherit the slots as own slots.

5. PRESENTING FMEA INFORMATION

From an ontological point of view, the deviation, its causes and consequences can be modeled in terms of causality relations. As shown in figure 4, a deviation is an event that is the beginning of a consequence (activity). A cause is an activity that causes deviation (event). This can also be represented by means of the tuples:

(beginning <activity> <event>)
(causes <activity> <event>)

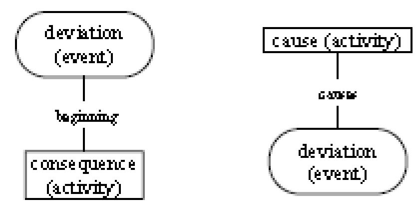

Figure 4 - Causality relations used to represent FMEA information

For example, the causes and consequences identified for a plant running in steady state in which all the reactants are in the reactor are different from those identified for the plant startup in which one or more of the reactants may not be in the reactor yet.

Often we are interested in finding not only the directed causes or consequences of the deviations but also the whole chain of events. To do this we have added the following rule expressed in the KIF language:

```
(forall (?act1 ?act2)
(=>
(and
(activity ?act1)
(activity ?act2)
(causes ?act1 ?event1)
(beginning ?act2 ?event1)
(causes ?act2 ?event2))
(causes ?act1 ?event2)
)
```

In English the rule can be interpreted as, if activity act1 causes event event1 and the beginning of activity act2 is event1 and act2 causes event2 then activity act1 causes event2.

Failure Mode and Effect Analysis

Page: 2 of 2	Documents:	Team Members:	Design/process/service
Date:	Es:	Design:	
Approval:	Technical dimensions:	Production:	Rigour Pin Machinary Test
	Control Plan:	Quality:	
	Contract:		

Actions	*Detect*	*Process Controls*	*Occurrence*	*Causes*	*Severity*	*Failure Effect*	*Failure Mode*	*Process Task*
Design preventive pins	3	Inspecting	2	Clock wise rotation	9	Unassembled	over scraping	Design preventive Pin
Design preventive pins	4	Inspecting	3	Wheel corrosion	8	Fitting problem	Pins problem	

Table 1 – FMEA example form

5 FMEA CASE STUDY

A fragment of the FMEA studies related to a Turbo Compressor in the utility unit of Bandar Imam Petrochemical Complexity is shown in Table 1. It can be seen from Table 1 that no information is provided in regards to the state of the plant that is assumed to exist before the deviation occurs. This may hamper the reuse of the results because the identified causes or consequences may not be valid if the plant is in another state or if the process is modified. Furthermore, it is difficult to reuse this information for the design of a different plant.

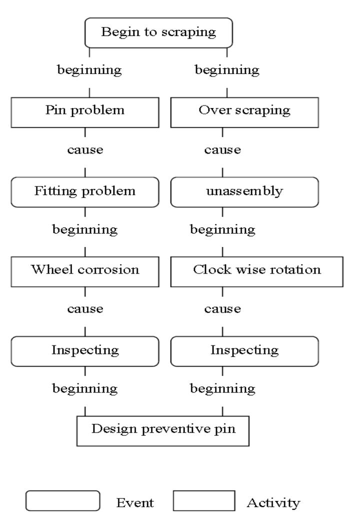

Figure 7 - Causality relations used to represent FMEA information

With the ontology, it is easy to express the relation between the plant under study, the state of the plant and the abnormal situations that were identified. Figure 7 shows a causality relation based on ontology concepts. The RDF/XML codes extracted by protégé considered as an input for JTP to get axioms or KIFs which are suitable for process designers and researchers to expand their point of view and opinions during design and engineering phases.

6 CONCLUSION

An ontology based on the standard ISO-15926 has been used in an effort to improve the representation of knowledge that is used and produced during Failure Mode and Effect Analysis (FMEA). Traditionally, the information that is used and produced during the FMEA 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 by protégé tool so that engineers can use more informative queries (instead of text based) to find relevant information during the safety analyses. On the other hand, OWL sacrifices expressivity for efficiency that nevertheless is an important requirement in an industrially deployed ontology.

REFERENCES

1. Berners-Lee, T., J. Hendler, and O. Lassila ; “The Semantic Web. Scientific American,” May, (2001).
2. Farquhar, A., R. Fikes, J. Rice ; “Tools for Assembling Modular Ontologies in Ontolingua,” Technical, Report KSL-97-03, Stanford University, KSL.
3. Gangemi A., N. Guarino, C. Masolo, A. Oltramari, and L. Schneider; “Sweetening Ontologies with DOLCE,” Proceedings of EKAW 2002. Siguenza, Spain (2002).
4. Genesereth, M. R. and R. E. Fikes; “KIF Version 3.0 Reference Manual,” [Online] Available: http://logic.stanford.edu/papers/kif.ps (1992).
5. Niles, I. and A. 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).
6. Sowa, J. “Knowledge Representation: logical, philosophical, and computational foundations” Brooks/Cole (2000).

7. W3C; "OWL Web Ontology Language Overview," W3C Recommendation, [Online] Available: http://www.w3.org/TR/owl-features/ (2004).

BIOGRAPHIES

Vahid Ebrahimipour, PhD.
Department of Industrial Engineering
College of Engineering
University of Tehran
Faculty of Engineering
P.O.: Box 14395-515 Tehran, Iran

e-mail: vebrahimi@ut.ac.ir

Vahid Ebrahimipour serves as an assistant professor at University of Tehran. He pursued his research related to Reliability Modeling under Prof. M.Xie's advice in National University of Singapore in 2007. He received his Ph.D. degree from the System Analysis Laboratory of Systems Engineering Department at Okayama University in Japan, in 2006. In 2002, he worked as a Research Assistant in the Environmental and Energy Engineering Laboratory of Tokyo University of A&T. He received his M.Sc. degree in Industrial Engineering from the University of Tehran in 2000. Also, he is an active technical program member of IEEE conference on IEEM, and member of Society of Instrument & Control Engineers (SICE) in Japan. His research interests include equipment dependability assessment, safety analysis, fault-tolerance control systems, fault recovery, information systems and security.

Kamran Rezaie, PhD.
Department of Industrial Engineering
College of Engineering
University of Tehran
Faculty of Engineering
P.O.: Box 14395-515 Tehran, Iran

e-mail: rezaie@rwtuviran.com

Kamran Rezaie is member of Faculty - Dept. Industrial Engineering, Faculty of Engineering, University of Tehran and managing director & member of board of TÜV NORD Iran joint Venture Co. also official and full authorized representative of TÜV NORD in IRAN, regional manager central Asia and chair of the board of TÜV Academy Iran–Germany. He received his PhD degree in Mechanical Engineering (Doktor der technischen Wissenschaften) from Technical University of Vienna and his MSc. in Mechanical Engineering (Diplom Ingenieur) Automotive & Transport system. He has been Conducting more than 270 Audits (ISO 9002 & ISO 14001) in different Industrial and Servicing companies in Iran and Conducted several training courses in fields of Quality Management and Quality Assurance (ISO 9000), Internal Auditor, Quality Techniques, Environmental Management (ISO 14000).

Samaneh Shokravi
Department of Industrial Engineering
College of Engineering
University of Tehran
Faculty of Engineering
P.O.: Box 14395-515 Tehran, Iran

e-mail: samans83@yahoo.com