



HAL
open science

A Formal Ontology for Industrial Maintenance.

Mohamed Hedi Karray, Brigitte Chebel-Morello, Nouredine Zerhouni

► **To cite this version:**

Mohamed Hedi Karray, Brigitte Chebel-Morello, Nouredine Zerhouni. A Formal Ontology for Industrial Maintenance.. Terminology & Ontology: Theories and applications, TOTh Conference 2011., May 2011, Annecy, France. pp.1-20. hal-00602182

HAL Id: hal-00602182

<https://hal.science/hal-00602182>

Submitted on 21 Jun 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Formal Ontology for Industrial Maintenance

Mohamed Hedi KARRAY*, Brigitte CHEBEL-MORELLO*, Nouredine ZERHOUNI*

* Automatic Control and Micro-Mechatronic Systems Department, 24, Rue Alain Savary,
25000 Besançon, France

Hedi.karray@femto-st.fr

Brigitte.morello@femto-st.fr

Nouredine.zerhouni@femto-st.fr

<http://www.femto-st.fr/fr/Departements-de-recherche/AS2M/Presentation/>

Abstract. The rapid advancement of information and communication technologies has resulted in a variety of maintenance support systems and tools covering all sub-domains of maintenance. Most of these systems are based on different models that are sometimes redundant or incoherent and always heterogeneous. In order to create a common and shared language, we are developing a domain ontology of industrial maintenance adopting METHONTOLOGY approach to manage the lifecycle development of this ontology that we have called IMAMO (Industrial MAintenance Management Ontology). The ontology developed can be exploited to ensure semantic interoperability and to generate new knowledge that supports decision making in the maintenance process.

1. Introduction

Industrial maintenance is a fundamental function in the business process and thus requires the development of computer systems [1]. It covers all domains of a business, from the plant and the equipment to be maintained, to the organization according to different strategies (preventive maintenance, predictive maintenance, corrective maintenance), to managing operators and material (handling, hoisting) and spare parts, to the computer-assisted diagnostic systems, to documentation management, etc. Various maintenance support systems and tools have begun to involve and to be essential for maintenance process management to cover all these sub-domains as CMMS, diagnostic support system, prognostic system, resource management systems as (ERP) and other systems. All these systems are currently based on different models that are usually complementary, but sometimes redundant, sometimes incoherent and always heterogeneous.

The principal problem confronted by these systems is to provide the means to move from coexistence to the interoperability and cooperation of different applications within the same environment.

Indeed, different researchers [2,3,4 and 5] are in agreement that Ontology engineering is recognized as the key technology to deal with this. Ontologies specify the semantics of terminology systems in a well defined and unambiguous manner [6], by formally and explicitly representing shared understanding about domain concepts and relationships between concepts. With Ontology intended meanings of terminologies and logical properties of relations are specified through ontological definitions and axioms in a formal language. The employment of a consistent Ontology is useful because maintenance support systems' users can have identical semantics for all concepts. All maintenance systems share the common Ontology of maintenance, so have an interoperable communication.

Additionally, we consider that a common neutral model is the most feasible solution in maintenance systems to enable data sharing or integration in heterogeneous applications. With a common Ontology model, it is possible for maintenance information to be created once, re-used and enriched during the equipment lifecycle. This reduces maintenance duration by eliminating the need to recreate data models repeatedly and increases maintenance quality by eliminating errors and using experience of previous maintenance activities. Ontology model organizes data and knowledge of a certain domain of interest (application) in a manageable manner. It should contain the definitions of all application objects (e.g. diagnostic, documentation) within that domain, as well as constraints and relationships between objects.

This, sharing of knowledge via the ontology allows to each system to exploit all the knowledge of the other systems. As well, the reasoning methods which can be applied on the ontology provide an added value to this knowledge while it can generate new knowledge that users cannot notice. This generated knowledge can be

exploited by all systems and then gives a surplus to maintenance actors usually in the aim to provide the right information in the right format to the Right people to do the Right things at the right time.

For these reasons, we provide in this work (in the scope of SMAC project¹) a domain ontology of maintenance domain to be shared and integrated in the maintenance platform as well as its integrated applications.

When a new ontology is going to be built, several basic questions arise related to the methodologies, tools and languages to be used in its development process [5]. From several methodologies developed to formalize ontologies creation we adopt METHONTOLOGY approach [9].

In the formalization phase, a UML ontological model is built in collaboration with maintenance experts; this ontology encloses 8 models classified as unavoidable for the maintenance process which are (1) the structural model presenting the equipment composition, (2) the functional model which characterizes different functionalities of the equipment and its components, (3) the dysfunctional model, (4) the event model which defines the triggering event launched after failures and/or degradation, and (5) the informational model which presents various resources (documents, human, software, tools...) related to the equipment and maintenance tasks, (6) the interventional model presenting concepts related to the intervention process, (7) the Strategy model presenting managerial aspects about maintenance strategy and contracts and (8) Process model presenting all technical, administrative and managerial process.

After that, to make the ontology operational, in the implementation phase of METHONTOLOGY we transform the UML model to PowerLOOM, a logic-based representation language for ontology presentation. Some proposed metrics [8] are then used to characterize the ontological model.

The remainder of this paper focuses the development process of IMAMO (Industrial MAintenance Management Ontology). Hence the rest of sections will be organized as follows. Section two is devoted to addresses the first three phases specification of METHONTOLOGY which are knowledge acquisition and conceptualization of the ontology. Section three is dedicated at the development of the formalization reusing and implementation of IMAMO. In section 3, the evaluation phase, a business-oriented approach based on cases of use is adopted to evaluate the ontology at the application level. Finally, we conclude the paper and we give some perspectives and guidelines to future work.

¹ <http://smac.univ-fcomte.fr>

2. IMAMO development process

Fernandez and al. in [9] assert that the ontology development process refers to the activities needed in order to build ontologies. In this context, METHONTOLGY has been proposed as a structured method to build ontologies. This methodology includes a set of activities, techniques, and deliverables to be produced after the execution of such activities using the corresponding techniques. The set of activities is {planning, specifying, acquiring knowledge, conceptualizing, formalizing, integrating, implementing, evaluating, producing documents, and maintenance}.

2.1 Specification

Domain	Industrial maintenance
Name	IMAMO: Industrial MAintenance Management Ontology
Date	2010
Conceptualized-by	Mohamed-Hedi Karray, Brigitte Morello, Thibault Bobyck
Implemented-by	Mohamed-Hedi Karray, Thibault Bobyck
Purpose	Ontology concerns most concepts of industrial maintenance when information about all technical, administrative and managerial activities and actions is required in maintenance information systems. This ontology can be used to ascertain decision making throughout the life cycle of maintenance activities from failure detection to intervention and repair.
Level of Formality	Formal
Scope	Structure of equipment to be maintained, spare parts, monitoring activity, failure detection, events, material resources, maintenance actors, technical documents, administrative documents, intervention, maintenance reports, equipment states, equipment life cycle....
Sources of Knowledge	Standards (AFNOR, MIMOSA..), projects, experts

TAB. 1 – *Ontology requirement specification document*

2.2 Knowledge acquisition

Creating an ontology is not a trivial task. It requires not only the skills of information technologies but also a significant knowledge in the modeled domain [10]. Hence, as mentioned in the specification document, we based our knowledge acquisition on standards, research projects and industrial experts in maintenance. Concerning standards, we adhere to the AFNOR norms and MIMOSA standards. The PROTEUS and PROMISE projects also serve as a base. Finally, we adopt the business expertise of various maintenance experts, managers and operators from different companies such as Cegelec SA France & Germany, Tornos (Switzerland), Peugeot (Belfort, France) and em@systemec (France). Various research works such as [11, 12 and 13] are also taken into account.

AFNOR defines maintenance as “the combination of all technical, administrative and managerial actions during the life cycle of an item intended to retain it in, or restore it to, a state in which it can perform the required function”. In concordance with this definition, Retour et al. [12] present the maintenance function as a set of activities grouped into two subsets: activities with technical predominance and activities with management predominance.

Consequently, the concepts which should be identified must cover all these layers and activities. To facilitate our identification of concepts, we identified all those concepts related to each layer presented above. Identification of the main concepts of each layer is based on the models of MIMOSA CRIS, the PROTEUS project and the PROMISE SOM and SMAC projects.

2.3 Conceptualization

In this phase, Gomer-Perez et al recommend a set of intermediate representations for conceptualizing a domain ontology of objects [9].

2.3.1 Glossary of terms and data dictionary

Hence, we begin the conceptualization of IMAMO by building the glossary of terms. The concepts are first classified in the glossary respecting the four layers identified by Rasovska et al. presented above. We then refined this list of concepts by deepening the first classification. The second classification goes further than the first by breaking each layer down into sub layers. We note that some concepts are redundant in different layers or sub-layers. This is done purposely in order to count all the concepts in each layer so as to obtain a clearer vision and a more precise identification. Next we edit the data dictionary based on the European norm NF EN 13306:2001 published by AFNOR. Due to limits of space we will not show these steps in three separate tables, but we will present only the data dictionary in the formalization phase in order to gain a better understanding of the different models of the ontology (see next sections).

IMAMO

We note, however, that IMAMO will be a generic ontology; different details can be neglected and left to users according to their needs. In this case, users may adapt, involve and maintain the ontology.

2.3.2 Concept classification trees

After this classification, we edit the concept classification trees. We notice that the domain is very broad; nevertheless, the ontology that we develop will not contain a lot of trees. This is due to the aim of obtaining a rich ontology with different types of relations and not a hierarchical ontology like taxonomy. *Is-a*, *is-component-of*, *has* and other verbs are the relations supported by the ontology. Figure 1 summarizes concept classification trees of resources in IMAMO (i.e. *is-a* relations).

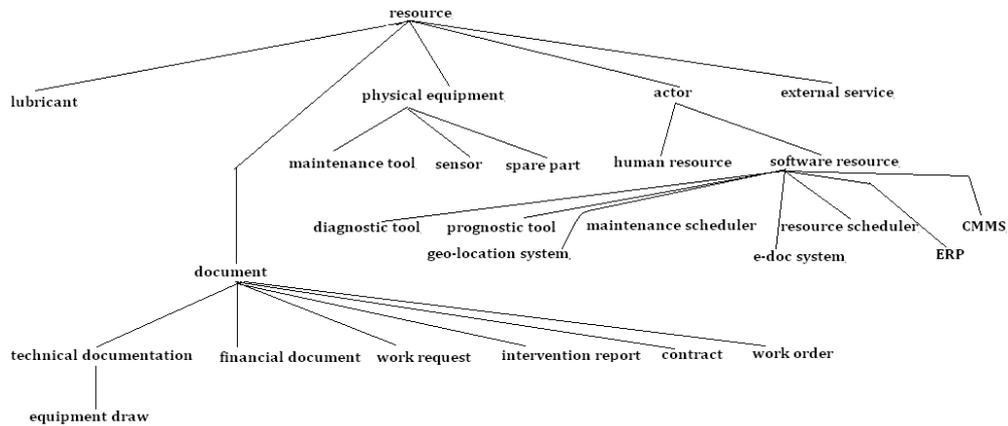


Fig. 1 – classification trees of resource in IMAMO

2.3.3 Edit rules

As mentioned above, rules will be left to user choice. We give here just some examples of rules that can be edited. We edit these rules by the description logics ALCQHI. Then rules can be translated and edited by the implementation language if it allows rules definition. Rules enrich the Ontology and allow more semantic reasoning as well as understandability.

For example thanks to the following defined rule the identification of critical components is possible without defining a new concept called critical component. The rule describe critical component as any physical equipment having the value of the propriety Functional-degree greats or equals to five.

CriticalComponent \equiv PhysicalEquipment \sqcap (\leq FUNCTIONNAL - DEGREE 5)

3. Formalization re-use and implementation phases

3.1 Formalization

The Unified Modeling Language (UML) would be a good candidate for representing ontologies and knowledge [14]. Knowledge expressed via UML is easily accessible for human comprehension. In an UML model, Knowledge can be changed easily due to the modular nature of object-oriented modeling. Also, new knowledge can be derived from UML models by reasoning about their contents [14]. With this standpoint, UML can be regarded as an appropriate candidate for knowledge representation. In this work we adopt the UML class diagram to formalize IMAMO. This choice is argued by the graphical expressivity and the semantics power of UML recommended in the various research works mentioned above. This fact facilitates the exchange between domain experts and the human understanding of the ontology. In the other hand, the ontology of the domain, although formalized independently of the methods of reasoning has a structure which depends on how acquired knowledge will be used for reasoning because experts deliver the knowledge adapted to their reasoning. Reasoning methods will be considered in the implementation phase.

We decompose the class diagram into seven models considered as inescapable for maintenance process according to the layers classification used in the conceptualization phase. These models are:

- (1) The structural model presenting the equipment composition; related to the equipment analyses layer.
- (2) The event model which define the triggering event launched after failures and/or degradation; related to the fault diagnosis and expertise layer.
- (3) The functional and dysfunctional model which characterizes different functionalities of the equipments and its components as well as which is related to the fault diagnosis and expertise layer.
- (4) The informational model which presents various resources (documents, human, software, tools, indicators...) related to the equipment and maintenance tasks as well as maintenance strategy and processes; related to resources management and maintenance strategy management layers.
- (5) The interventional model presenting concepts related to the intervention process.
- (6) The strategy model presenting managerial aspects about maintenance strategy and contracts.

(7) The process model presenting all technical, administrative and managerial process.

Below we present the tow first models with the data dictionary containing definitions of the modeled concepts.

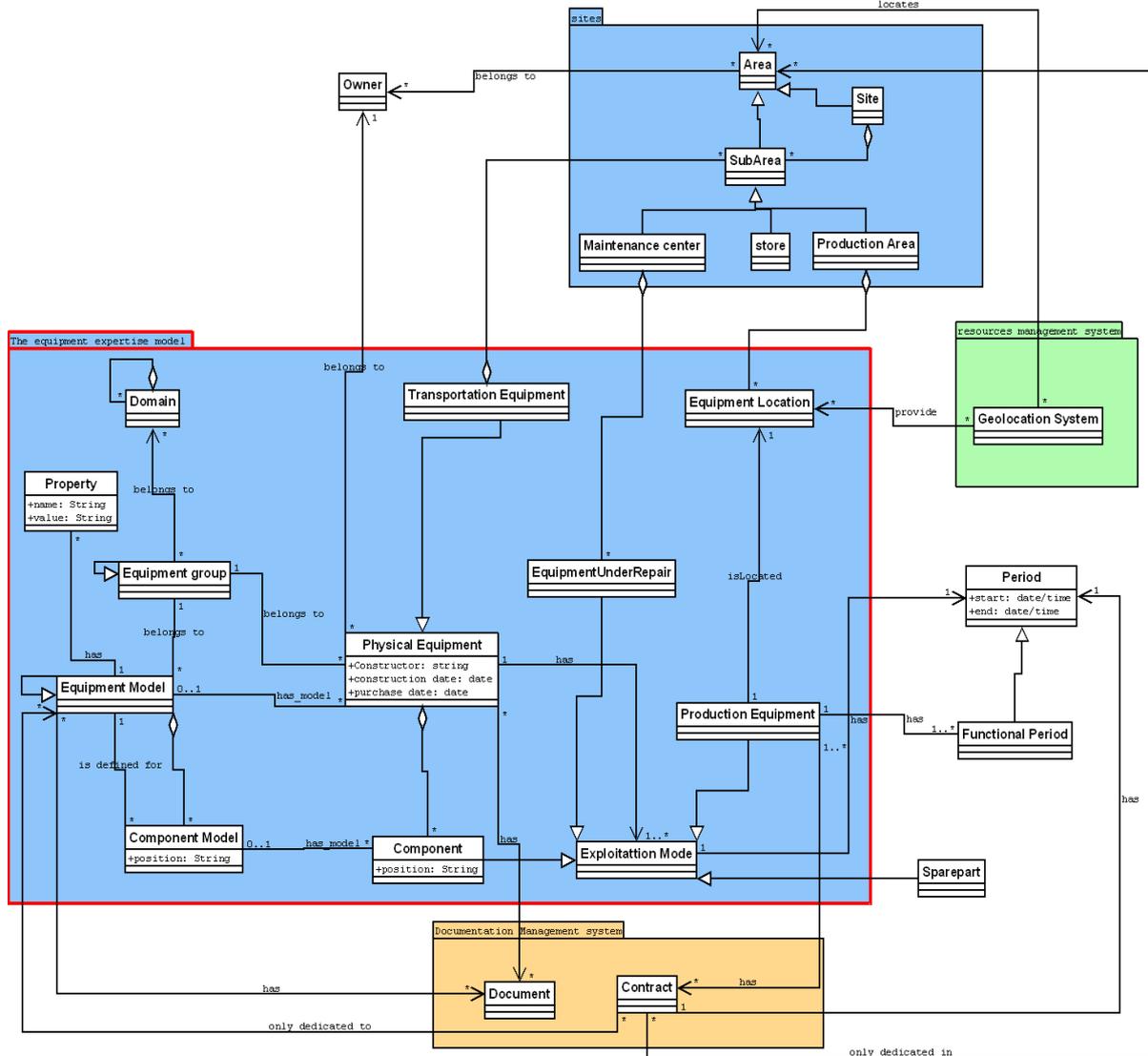


Fig 2– Structural model

Concept Name	Synonyms	Description
domain;		A particular field of knowledge or expertise (e.g. hydraulic).

physical equipment;	Asset Physical- product Machine Device Item	Tangible, instantiated, serialized object, component, device, subsystem, functional unit, equipment or system that can be individually considered to be maintained. A physical equipment may be an entire facility, an entire functioning platform (such as an CH-47 Tail Number XYZ helicopter), or a component piece of equipment, such as a specific instance of a bearing.
transportation equipment;		Particular Physical Equipment conveyance. A conveyance which may contain one or more area (s) of production, a set of maintenance teams, and a set of stores. For example: A fishing vessel off the coast has its own decomposition (motor allowing it to move...) as well a production area that cleans the fish and freeze.
maintenance tool;		Particular physical equipment that used as tool for performing the maintenance activities. This type of physical equipment has to be maintained also.
equipment model;	As-designed-product Model	Conceptual view of physical composition of the equipment. It is composed by the various components models of the components composing a physical equipment.
component model;		Conceptual view of a component (e.g. model of electrical motor).
Component mode;		Component is an exploitation mode that can be played by a Physical Equipment. It has the particularity to be within superior physical equipment (e.g. motor3X57H).
exploitation Mode;		Abstraction of a role played by equipment. It presents the state of exploitation that can take a physical equipment. It can be exploited as a component, a production equipment, a spare part or be under repair.
equipment under repair mode;		Specific exploitation mode affected to a Physical Equipment while it is during repair or located in a maintenance center to be repaired.
production equipment mode;		Specific exploitation mode affected to a Physical Equipment while it is exploited in a production tasks and/or located in a production area.
Spare part mode;		Specific exploitation mode affected to Physical equipment intended to replace corresponding physical equipment in order to restore the original required function of the physical equipment. Generally, it is located in a store.
equipment location;		Position of a physical equipment in a production area (to locate and track the positions of equipment).
area;		Particular geographical region (to manage multisite).
sub area;		Region that makes up part of an area.
site;		Place or setting of something. An area or plot of ground with defined limits on which a building, project, park, etc., is located or proposed to be located.
maintenance	Maintenance	Specific area to do maintenance tasks.

		generally retrieved from sensors into digital values for processing.
condition;		Environmental or functional requirement defined to supervise (monitoring task) a specific physical equipment or a place (e.g. site) by the use of sensors and data acquisition systems.
triggering event;		Something that happens at a given physical equipment and time that triggers a specific process of maintenance which is work request process.
alarm;		Type of triggering event launched from a data acquisition system indicating that there is a measure from a sensor violating some conditions about a specific equipment or environment.
improvement request;		Triggering event about a specific or general asking for the improvement of a physical equipment. Knowing that an improvement is the combination of all technical, administrative and managerial actions, intended to ameliorate the dependability of a physical equipment, without changing its required function.
event observed by user;		Type of triggering event about a dysfunction on a physical equipment observed by user which is a human resource.
notification;		Type of triggering event informing about future events as planned maintenance or the prognostic RUL.
prognostic;		Type of notification composed by the health status at a future time and the remaining useful life (RUL) of a physical equipment. It is the output of the prognostic tool.
Prognostic tool;		Software too or system allowing to predict and estimate the rest of time to failure and the risk of subsequent existence of one or more failure modes with a confidence level which is a value indicating the degree of certitude that the prognosis is correct.
Maintenance scheduler;		Type of software resource that allows planning, allocation of significant amount of time, and high degree of coordination between different departments, and is typically initiated through a work order. It is considered as a software resource.

Tab3– *Data dictionary of event model*

3.2 Reusing: integration and merge

While ontologies are built to be reused, Ontology reuse is one of the important issues in ontology construction. According to Pinto et al there are two different reuse

IMAMO

processes [15]: merge and integration. Both of these reuse processes are included in the overall process of ontology building. Merge is defined as the process of building ontology in one subject reusing two or more different ontologies on that subject [15]. Otherwise, in an integration process source ontologies are aggregated, combined, assembled together, to form the resulting ontology, possibly after reused ontologies have suffered some changes, such as, extension, specialization or adaptation. We should note that both processes are adopted in this work.

To create IMAMO we started from models developed in PROTEUS project, we involved a first version of maintenance ontology published in [16]. This ontology, is composed by 62 concepts and 70 relations integrating the main concepts used in PROTEUS. Then, in the scope of SMAC project [17] we have done a mapping between this ontology with the PROMISE model. As result of this work, Matsokitis and Karray [18] proposed an involved version of this ontology by orienting it to maintenance field when integrating some concepts related to the maintenance area included in the MOF phase of PLM. This ontology called SMAC-model is formalized by UML and implemented with OWL-dl via PROTÉGÉ tool.

Then, we come back to maintenance scope, we integrating some concepts from SMAC-model related to the life cycle of equipment to take into account the beginning of life part concerning the design phase as well as the middle of life phase by tracking all the events and health states of the equipment and the end of life by the calculation of indicators supporting decision for reuse and disassembly. An example of some reused concepts integrated in IMAMO is presented in the next table. In addition, as mentioned above, MIMOSA-CRIS is considered as the reference of the domain, for this reason, when creating IMAMO we taken account classes used in this model. A mapping between MIMOSA-CRIS and IMAMO is done, some labels of reused concepts are changed but the alignment between these concepts is added as equivalence or subsumption rules after the mapping between the two ontologies.

IMAMO Concepts	From which ontology?
Physical equipment	PROMISE: <i>Product</i> -- MIMOSA-CRIS: <i>Asset</i>
Equipment Model	MIMOSA-CRIS: <i>Model</i> -- PROMISE: <i>As-designed-product</i>
Equipment group	MIMOSA-CRIS: <i>Asset type</i> -- PROMISE: <i>product group</i>
Site	MIMOSA-CRIS: <i>site</i> – SMAC-Model: <i>Location site</i>
Triggering event	MIMOSA-CRIS: <i>Event type</i> -- PROMISE: <i>Event</i>
Measure	MIMOSA-CRIS: <i>Measurement Event</i> -- PROMISE: <i>Field Data</i>
Equipment location / Geo-location system	MIMOSA-CRIS: <i>Geoposition</i>
Alarm	MIMOSA-CRIS: <i>Alarm type</i> – SMAC-Model: <i>Alarm</i>
Process pattern	MIMOSA-CRIS: <i>Work Management Type</i> -- SMAC-model: <i>process</i>

Table 10- Example of reuse table

In the other hand, other integration is possible as of the functional ontology proposed by Mizoguchi et al [19]. This ontology can be integrated under the concept Function The ontology of faults presented by Kitamura et al [20] can be integrated to involve the dysfunctional model of IMAMO. As well as, task ontologies as the one presented in [21] by Ikeda et al can be integrated under the concept Task.

3.3 Implementation

We translate the UML model of IMAMO to PowerLoom. We chose to work with PowerLoom version 3.2.0 despite version 4.0 is actually available because the version 3.2.0 is a stable version unlike to the forth version which is a beta version yet.

In addition, we note that a PowerLoom exporter for the Protégé editor has been implemented. It can write ontologies using the Protégé frame language in PowerLoom, either fully native or with support for the system concepts from Protégé. Moreover The PowerLoom GUI (or knowledge editor) a Java-based graphical client for PowerLoom is now standardly included and available with PowerLoom starting with version 4.0 [22].

We present in the following a part of the structural model of the equipment implemented by PowerLOOM. Each UML class is translated into a PowerLOOM concept using "DEFCONCEPT" command. Associations and attributes of classes are translated into PowerLOOM relation or function using "DEFFUNCTION" and "DEFRELATION" commands.

```
(DEFMODULE "/PL-KERNEL-KB/PL-USER/ONTOLOGIE-MAINTENANCE"
:DOCUMENTATION "Module for Maintenance"
:INCLUDES ("PL-USER"))

(IN-MODULE "/PL-KERNEL-KB/PL-USER/ONTOLOGIE-MAINTENANCE")
(IN-DIALECT :KIF)
(DEFCONCEPT COMPONENT)
(DEFCONCEPT PHYSICAL-EQUIPMENT)
(DEFRELATION PHYSICAL-EQUIPMENT-ID ((?C PHYSICAL-EQUIPMENT)
(?ID STRING)))
(DEFRELATION PHYSICAL-EQUIPMENT-TYPE ((?C PHYSICAL-EQUIPMENT)
(?TYPE STRING)))
.....
(DEFCONCEPT EQUIPMENT-MODEL)
(DEFRELATION EQUIPMENT-HAS-TOP-MODEL ((?E PHYSICAL-EQUIPMENT)
(?MG EQUIPMENT-MODEL)))
(DEFRELATION EQUIPMENT-MODEL-INHERITS ((?MG1 EQUIPMENT-MODEL)
(?MG2 EQUIPMENT-MODEL)))
(DEFRELATION EQUIPMENT-COMPONENT-COMPOSED ((?E PHYSICAL-
EQUIPMENT) (?C COMPONENT)))
```

IMAMO

```
(DEFRELATION EQUIPMENT-COMPOSED ((?E PHYSICAL-EQUIPMENT) (?COM  
PHYSICAL-EQUIPMENT)))
```

4. Evaluation and maintenance

The aim of our evaluation is to highlight the added value of IMAMO to maintenance systems and actors as well as in order to provide new users with sufficient information to promote the exploit of this ontology (the extent of the maintenance domain coverage). Another aim of this evaluation is to focus the weak points of this ontology to facilitate maintenance and evolution tasks.

Our approach of evaluation is composed on two main steps [23]: the quality of the formalization model and the business oriented evaluation containing the evaluation of applicability, efficiency and simplicity for interoperability.

4.1 Quality of the model

In accordance with Tartir et al [24], assessing the quality of an Ontology is important for several reasons including allowing the Ontology developer to automatically recognize areas that might need more work, allowing the Ontology user to know what parts of the Ontology might cause problems. Different dimensions are available to assess the quality of an ontology. We are interesting to quality metrics presented by [24]. We use metrics of schema evaluation to evaluate the success of the Ontology UML model of real-world domain of maintenance. How classes are organized, the depth, the richness, breadth, and height balance of the Ontology schema inheritance tree can play a role in a quality assessment.

To understand used metrics and bellow discussion please refers to [24]. We notice that the UML class diagram of IMAMO contains 187 Relations (P), 103 Concepts (classes) (C), 60 Subclasses (SC) and 40 Attributes (att).

The first metric is called Relationship Richness (RR). This metric reflects the diversity of relations and placement of relations in the Ontology. The RR of our ontological model is 75%. It exceeds largely the average. This means that our Ontology is not a hierarchic Ontology. It is not just a hierarchy of subclasses but it maintains a balance between inheritance relations and role associations. This is due to the inclusion of domain concepts and proves that the ontological model is business oriented and responds to business needs of maintenance.

The second used metric is called Attribute Richness (AR). The number of attributes that are defined for each class can indicate both the quality of Ontology design and the amount of information pertaining to instance data. In general we assume that the more slots that are defined the more knowledge the Ontology conveys. The AR of IMAMO is 0.38. This result shows the poverty of the ontological model in terms of attributes. The result gives as average 0.38 attributes per concept which is very low. This is known to us from the beginning because when constructing the

model we focus our interest on the concepts presentation and not its details information by the identification of attributes. Although we know that these details are very interesting for the quality and richness of the Ontology, this is a very difficult to implement because the Ontology must use expressive and unambiguous terms. In this context, it is a difficult task to be generic more than in the context of concepts. However, in our future work, collaborations with business experts are envisaged to identify attributes that may be generic for related maintenance concepts (e.g. details of classes of the UML ontological model).

The third used metric is called Inheritance Richness (IR). This measure describes the distribution of information across different levels of the Ontology's inheritance tree or the fan-out of parent classes. This is a good indication of how well knowledge is grouped into different categories and subcategories in the Ontology. This measure can distinguish a horizontal Ontology from a vertical Ontology or an Ontology with different levels of specialization. The IR obtained of IMAMO is 0.58. This result is near to the average. This shows that in the context of knowledge details, our Ontology maintains the balance between generality and explicitness. The ontological model is hybrid; it is neither vertical nor horizontal. We consider this result as a reached target, because our first goal was to build a generic ontological model to the maintenance domain, but at the same time not very light, so that it covers as possible all maintenance aspects (concepts).

4.2 Business evaluation of the ontology

4.2.1 Application use case

The application and tests IMAMO was done on the pallet transfer system. It represents a flexible production system. It is composed of five robotized working stations which are served by a transfer system of pallets organized into double rings (internal and external). Each station is equipped with pneumatic actuators (pushers, pullers and indexers) and electric actuators (stopper) as well as a certain number of inductive sensors (proximity sensors). An inductive read/write module allows to identify and locate each pallet and to provide information relative to required operation in a concrete station. The working station is situated on the external ring and contains pneumatic and electric actuators (puller, pusher, indexer, and stopper) as well as inductive sensors.

The following PowerLOOM code specifies how we can assert the `PHYSICAL-EQUIPMENT` `SISTRE`, the `EQUIPMENT-MODEL` `PLATEFORME`, as well as associations `PHYSICAL-EQUIPMENT-CONSTRUCTOR` "Bosch" of `SISTRE`, the top-model `EQUIPMENT-HAS-TOP-MODEL` of `SISTRE` which is `PLATEFORME` and the association `EQUIPMENT-COMPONENT-COMPOSED` to describe the composition of the physical equipment `SISTRE`:

IMAMO

```
(ASSERT (PHYSICAL-EQUIPMENT SISTRE))
(ASSERT (PHYSICAL-EQUIPMENT-CONSTRUCTOR SISTRE "Bosch"))
(ASSERT (EQUIPMENT-MODEL PLATEFORME))
(ASSERT (EQUIPMENT-HAS-TOP-MODEL SISTRE PLATEFORME))
(ASSERT (EQUIPMENT-COMPOSED SISTRE ROBOT))
.....
(ASSERT (EQUIPMENT-COMPONENT-COMPOSED CONVOYEUR ENTRETOISE))
(ASSERT (EQUIPMENT-COMPOSED CONVOYEUR COURROIE))
```

The java API of PowerLOOM gives the possibility to query the knowledge base (e.g. assertion of the Ontology). In this example we ask PowerLOOM engine to list all Physical equipment. The given answer is not just the physical equipment SISTRE but its composition. In our Ontology we defined that a Physical-equipment can be composed by Physical-equipments.

```
PL-USER |= (load "ontologie-maintenance.plm")
PL-USER |= (in-module "ONTOLOGIE-MAINTENANCE")
ONTOLOGIE-MAINTENANCE |= (RETRIEVE ALL (PHYSICAL-EQUIPMENT
?PE))
```

```
There are 28 solutions:
#1: ?PE=COURROIE
#2: ?PE=CONVOYEUR
#3: ?PE=DETECTEUR
#4: ?PE=ACTIONNEUR
#5: ?PE=TEST
.....
#28: ?PE=TAP-INT
```

The second example show how PowerLOOM reasoning engine take account inheritance (subclasses). We launch a query to list all ACTIONNEUR and then we launch a query to ask only ACTIONNEUR-ELECTRIQUE. The first answer includes the answer of the second query due the sub-class relation.

```
ONTOLOGIE-MAINTENANCE |= (RETRIEVE ALL (ACTIONNEUR ?A))
There are 8 solutions:
```

```
#1: ?A=INDEXEUR
#2: ?A=TIREUR
#3: ?A=POUSSEUR
#4: ?A=S5
#5: ?A=S4
#6: ?A=S3
#7: ?A=S2
#8: ?A=S1
```

```
ONTOLOGIE-MAINTENANCE |=
(RETRIEVE ALL (ACTIONNEUR-ELECTRIQUE ?AE))
There are 5 solutions:
```

```
#1: ?AE=S5
#2: ?AE=S4
#3: ?AE=S3
#4: ?AE=S2
#5: ?AE=S1
```

We provide a second example concerning two maintenance systems S1 and S2. The first uses the MIMOSA-CRIS model, and the second uses IMAMO. The equipment SISTRE fails. S1 sends an intervention request to S2. S2 needs all information about the equipment composition and previous intervention on the equipment. An equipment is modeled by the concept ASSET in the MIMOSA-CRIS model. In IMAMO an equipment is defined by the concept PHYSICAL-EQUIPMENT. We show a sequence of PowerLOOM examples to demonstrate how the reasoning engine can solve this problem and how it manages to get the information expected (needed) by S2.

Firstly we ask the PowerLOOM system of the Ontology used in S2 to list all equipment which it knows, after that if it knows that SISTRE is an equipment:

```

ONTOLOGIE-MAINTENANCE |= (RETRIEVE ALL (PHYSICAL-EQUIPMENT
?PE))
There are 3 solutions:           #2: ?PE=TEST01
#1: ?PE=TEST02                 #3: ?PE=TEST

ONTOLOGIE-MAINTENANCE |= (ASK (PHYSICAL-EQUIPMENT SISTRE))
UNKNOWN
ONTOLOGIE-MAINTENANCE |= (ASK (not (PHYSICAL-EQUIPMENT
SISTRE)))
UNKNOWN

```

So the system of S2 doesn't know if SISTRE is an equipment. To ensure an understandable exchange between S1 and S2, it suffices to add in the knowledge base an equivalence rule defining ASSET as a concept equivalent to PHYSICAL-EQUIPMENT. This will allow to S2 to get all information about all concepts having as type ASSET.

```

(DEFCONCEPT PHYSICAL-EQUIPMENT ((?A ASSET)))
(DEFCONCEPT ASSET ((?PE PHYSICAL-EQUIPMENT)))
After that we assert that SISTRE is an ASSET
(ASSERT (ASSET SISTRE))

```

When we ask if ASSET is physical equipment the reasoning engine answer true:

```

ONTOLOGIE-MAINTENANCE |= (ASK (PHYSICAL-EQUIPMENT SISTRE))
TRUE

```

So, when the S2 system considers that ASSET as PHYSICAL-EQUIPMENT, it can obtain composition information of SISTRE from S1 while the two systems shares the same knowledge base.

IMAMO

```
ONTOLOGIE-MAINTENANCE |= (retrieve all (EQUIPMENT-COMPOSED  
SISTRE ?x))
```

There are 4 solutions:

#1: ?X=ROBOT

#3: ?X=DETECTEUR

#2: ?X=ACTIONNEUR

#4: ?X=CONVOYEUR

5. Conclusion

Maintenance domain contains a variety of maintenance support systems and tools covering all its sub-domains. Contrariwise the largest part of these systems are based on different models that are sometimes incoherent and always heterogeneous. To respond this problem, we propose to create a domain ontology for industrial maintenance that we called IMAMO (Industrial MAintenance Management Ontology).

In this paper we presented the development lifecycle of IMAMO. From various exiting methodology of ontology construction, we adopted METHONTLOGY approach to manage the lifecycle of our maintenance ontology.

To develop this ontology, we are based on different standards and previous projects elaborated in the domain of industrial maintenance. The formalization of the ontology is done via UML class diagram which provides a rich conceptualization of the complex domain of maintenance accompanied by the data dictionary. To be operational, a logic-based representation language called PowerLOOM characterised by its power and efficient reasoning engine is used to implement IMAMO. Afterwards, different approaches are applied to evaluate the proposed ontology in two levels. The first one is to characterize the ontological model and the functional quality of the ontology. The second level is applied in order to show how IMAMO can resolve semantic interoperability problem.

Concerning shortage of this work needed to be addressed in future work, more collaboration with maintenance experts from industrial and academic communities to evolve and enrich the proposed ontology is needed. In addition, as perspectives, we aim set up of a web site containing documentation about the ontology and allowing the possibility to make maintenance and evolution activities.

Acknowledgment

This work was carried out and funded in the framework of SMAC project (Semantic-maintenance and life cycle), supported by Interreg IV programme between France and Switzerland.

References

1. Liyanage, J.P. and Kumar, U., Towards a value-based view on operations and maintenance performance management, *Journal of Quality in Maintenance Engineering*, Vol. 9, pp. 333–350
2. Heiler S. (1995) Semantic Interoperability. *ACM Computing Surveys (CSUR)*.
3. Riichiro Mizoguchi (2004). "Tutorial on ontological engineering". In: *New Generation Computing*. Ohmsha & Springer-Verlag, 22(2):198-220.
4. Yang Q Z & Zhang Y. (2006) Semantic interoperability in building design: Methods and tools. *Journal of Computer-Aided Design* 38 pp 1099–1112. Amsterdam: Elsevier Science Publishers.
5. Obrst, L. (2003). *Ontologies for Semantically Interoperable Systems*. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 366–369, New York, NY, USA. ACM Press.
6. Guarino N. (1998) *Formal Ontology and information systems*. Formal Ontology and information systems. IOS Press.
7. O. Corcho, M. Fernandez-Lopez, and A. Gomez-Perez, *Methodologies, tools and languages for building ontologies: Where is their meeting point?*, *Data & Knowledge Engineering*, Vol. 46, 2003, 41-64.
8. Tatir, Samir; Arpinar, I. Budak. *Ontology Evaluation and Ranking using OntoQA*. *Int. Conf. on Semantic Computing(ICSC)*, 2007
9. Gómez-Pérez, A.; Fernández, M.; de Vicente, A. (1996). *Towards a Method to Conceptualize Domain Ontologies*. *Workshop on Ontological Engineering. ECAI'96*. Budapest. Hungary. PP: 41-52.
10. FRANKOVIČ, Baltazár - BUDINSKÁ, Ivana. *The role of ontology in building of knowledge systems for industrial applications*. In *4th Slovakian - Hungarian Joint Symposium on Applied Machine Intelligence : proceedings*. ISBN 963-7154-44-2. s. 15-25.
11. Kaffel H. (2001) *La maintenance distribuée: concept, évaluation et mise en oeuvre*. Phd thesis, Université Laval, Quebec.
12. Retour D., Bouche M. et Plauchu V., *Où va la maintenance industrielle*, *Problèmes Économiques*, No. 2.159, pp. 7-13, 24 Janvier 1990.
13. Rasovska I, Chebel-Morello B & Zerhouni N. (2004) *A conceptual model of maintenance process in unified modeling language*. *Proceedings at 11 th IFAC Symposium on Information Control Problems in Manufacturing 2004 (INCOM)*
14. Stephen Cranefield, *Networked Knowledge Representation and Exchange using UML and RDF*, *Journal of Digital Information*, Vol 1, No 8 (2001).
15. H. Sofia Pinto, A. Gómez-Pérez, J. P. Martins. *Some Issues on Ontology Integration*. In *Proc. of IJCAI99's Workshop on Ontologies and Problem Solving Methods: Lessons Learned and Future Trends*, 1999.
16. Karray M H, Morello-Chebel B, Zerhouni N, "towards a maintenance semantic architecture", *Proceedings of The Fourth World Congress on Engineering Asset Management (WCEAM 2009)* Athens.
17. www.smac.fr

IMAMO

18. Matsokis, A., Karray, M.H., Morello-Chebel, B. and Kiritsis, D. (2010a). An Ontology-based Model for providing Semantic Maintenance. Proceedings of the 1st IFAC workshop on Advanced Maintenance Engineering, Services and Technology (A-MEST'10); 1-2 July 2010, Lisbon, Portugal.
19. Yoshinobu Kitamura and Riichiro Mizoguchi, "Functional Ontology for Functional Understanding", International Workshop on Qualitative Reasoning (QR-98), Cape Cod, USA, May 26-29, AAAI Press, pp.77-87, 1998
20. Yoshinobu Kitamura and Riichiro Mizoguchi, "An Ontological Analysis of Fault Process and Category of Faults", Proc. of Tenth International Workshop on Principles of Diagnosis (DX-99), pp.118-128, June 8-11 1999 DX 99 Loch Awe, Scotland 118
21. Ikeda, M., Seta, K., and Mizoguchi, R. (1997) Task Ontology Makes It Easier To Use Authoring Tools. Proc. of IJCAI-97, Nagoya, Japan, 342-347.
22. <http://www.isi.edu/isd/LOOM/PowerLoom/>
23. Brank, Janez; Marko Grobelnik; Dunja Mladenić. 2005b. A survey of ontology evaluation techniques. SiKDD05.
24. Tatir, Samir; Arpinar, I. Budak. Ontology Evaluation and Ranking using OntoQA. Int. Conf. on Semantic Computing(ICSC), 2007

Résumé

Le progrès rapide des technologies de l'information et de la communication a donné lieu à une variété de systèmes d'aide à la maintenance et des outils couvrant l'ensemble des sous-domaines de la maintenance. Les pluparts de ces systèmes sont basés sur différents modèles qui sont parfois redondantes ou incohérentes et toujours hétérogène. Afin de créer un langage commun et partagé, nous avons développé ontologie de domaine pour la maintenance industrielle en adoptant l'approche METHONTOLOGY pour gérer le développement du cycle de vie de cette ontologie que nous avons appelé IMAMO (). L'ontologie développée peut être exploitée pour assurer l'interopérabilité sémantique et de générer de nouvelles connaissances qui supporte la prise de décision dans le processus de maintenance