Ontology for complex railway systems application to ERTMS/ETCS system

Olimpia Hoinaru, Georges Mariano, Christophe Gransart

To cite this version:

Olimpia Hoinaru, Georges Mariano, Christophe Gransart. Ontology for complex railway systems application to ERTMS/ETCS system. FM-RAIL-BOK Workshop in SEFM'2013 11th International Conference on Software Engineering and Formal Methods, Sep 2013, Spain. 6p. hal-00912733

HAL Id: hal-00912733

https://hal.science/hal-00912733

Submitted on 2 Dec 2013

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.

Ontology for complex railway systems application to ERTMS/ETCS system

Olimpia Hoinaru†, Georges Mariano†, Christophe Gransart‡
* Université Lille Nord de France
French Institute of science and technology for transport,
development and networks (IFSTTAR)
[olimpia.hoinaru|georges.mariano|christophe.gransart]@ifsttar.fr
† ESTAS
Évaluation des Systèmes de Transports Automatisés et de leur Sécurité
‡ LEOST
Laboratoire Électronique Ondes et Signaux pour les Transports

***Abstract*—We present hereafter the experimental work of building an ontology of the European Rail Traffic Management System (ERTMS) domain. ERTMS is a railway complex control system defined on the basis of publicly available specification documents, the System Requirement Specification (SRS). We will describe the methodology that we used to define an initial structure for an ERTMS ontology. The main goal of this work is to supply a first formalization of the ERTMS knowledge in order to provide the basis of a later development process i.e validating the specifications, developing the software/hardware components and finally validating the system.**

***Index Terms*—Ontologies, ERTMS/ETCS, railway systems**

I. Introduction

ERTMS stands for the European Rail Traffic Management System. This is a European standard for the process control system and signalling and new lines for the replacement of existing systems for conventional lines. ERTMS contains two basic elements:

GSM-R (Global System for Mobiles - Railway): the communication component containing a voice communication network between vehicle drivers and line controllers. It provides routing and portability for ETCS data. It is based on the GSM standard public with specific features for railways.

ETCS (European Train Control System): the signalling system component that includes control movement authorities, automatic train protection and interface with the interlocking.

Developing such a complex structure is, of course, a real challenge. Only by considering the development of the corresponding software, we can observe on the figure 1 the general evolution of the technologies employed.

Roughly speaking, in the past, the challenge was to define a method to derive machine code from documentation (this documentation coming from the informal, and sometimes implicite "knowledge" of the system to be developed). To answer this challenge, (countless) modelling methods were defined and are now available. Thus we may now affirm that the code is correct because it corresponds to previously established models (whether they are formal or not). What we will explore hereafter is the concern that the question is now : How can we provide good models (preferably formal ones)? By doing this, we completely follow the paradigm stated in [1] : "Before software can be designed we must understand the requirements. Before requirements can be finalised we must have understood the domain". But where Dines Bjorner uses pure formal logic to tackle generic sample problems, we will experiment the use of ontological technologies (conceptualization, formalization, reasoning) to tackle a real and complex system.

II. General goal(s)

The work presented in this article is situated at the intersection of several domains i.e. knowledge management and Web semantics, knowledge representation and formalization, as well as system modelling. The knowledge of the ERTMS domain is considered and formalized for understanding and reuse issues.

Several methods (models) can be used to capture the different aspects of a railway complex system. Based on the fact that the same concept can have different meanings in different domains, the need for specification of these semantic differences was felt.

The ERTMS ontology aims at modelling and formalizing the System Requirements Specification documents of the ERTMS. These documents are written in natural language. The aim of this ontology is the formalization of these specifications in order to obtain a data structure that can be reusable in the framework of other research in the ERTMS field. A module of this ontology is the OSI (Open Systems Interconnection) [2] model and another one concerns the application of the OSI model to the ERTMS/ETCS subsystem dealing with the data transmission by means of radiocommunication.

III. Elaborating ontologies

Ontologies are formal representations of knowledge of a certain domain. Several definitions of the term "ontology" have been provided. [3] poses that "an ontology is an explicit specification of a conceptualization". According to the same

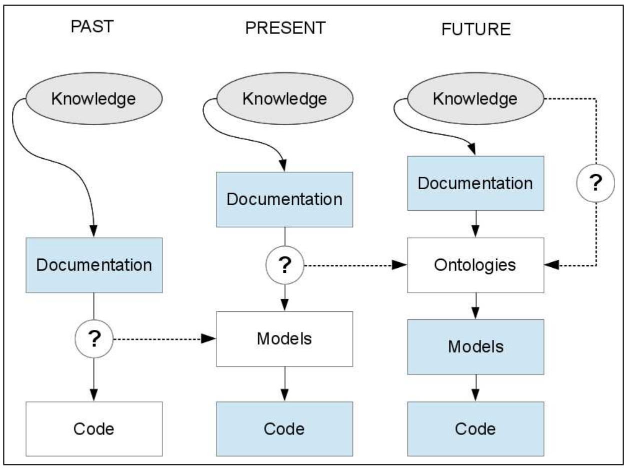

Fig. 1. Ontologies for software-based systems

author ”the term is borrowed from philosophy, where an ontology is a systematic account of Existence”.

There are four types of information allowing us to precise what is that we represent in an ontology. These are the type of ontology (domain ontologies, generic ontology, ontology of a method of solving a problem, application ontology and representation ontology), the properties, the ”is-a” relation and the other relations [4].

The knowledge of a domain is formalized using several notations with the aim of regrouping and creating a formal structure of the concepts of this domain into a web of knowledge.

We chose an ontology creation tool using the Web Ontology Language (OWL), i.e. the Protégé tool. Protégé-2000 was developed by Mark Musen’s group at Stanford Medical Informatics. In this environment, concepts are formalized as classes together with their several types of properties and the relations between them. The so-called ”rules” are created for the purpose of modelling requirements and certain ”behaviors” of the system.

In the railway domain, documents describing the System Requirements Specifications were issued with the specific aim of explaining and clarifying the usage of a part of the terms/concepts used in this domain, and of the system itself.

A. *Approaches*

This paragraph presents some of the ontology development methodologies existing. ”Methontology” is the term used to describe one of the methodologies for creating an ontology. It is among the more comprehensive ontology engineering methodologies as it is one for building ontologies either from scratch, reusing other ontologies as they are, or by a process of re-engineering them.

But methontology is not the only methodology of creating ontologies. Other methodologies like, for example, the corpus-based methodology exist. In this case, the ontology is derived from documents provided in natural language that can also contain diagrams, flow charts, or tables. It is the case of the ERTMS ontology whose creation we are presenting in this study.

- [5] is a publication dealing precisely with this subject-matter. The authors describe here the reasons that can lead one to develop an ontology i.e. the usage of this kind of structure, its definition, several types of methodologies, as well as the composition and structure of an ontology. We found this article particularly interesting for its explicitness and pedagogical style. The example taken is a test ontology created by the Protege developers, a wine ontology.

IV. Global view of the proposed ERTMS ontology

A. *The chosen method*

Our ontology is based on normative documentation, i.e. the System Requirements Specification [6] documents provided by the European Railway Agency (ERA). Other related documents are the ”ERTMS Glossary” and the ”ETCS Implementation Handbook” published by International Union of Railways (UIC). This is an ontology created as a semantic model and module extracted from the below mentioned documents. The extraction is based on the study, the comprehension of these documents, and on the transposition of the information conceptualized in the same documents. All this is being carried out manually by (some of) the authors of this article and not performed automatically as some software can do. As the study of these SRS within the framework of this research is at its beginnings, we chose to start it manually for a better usage of the comprehension of the human understanding. A perspective of this study is the automation of the information extraction from the SRS and other documents. This ontology is a way of formalizing the information provided by these documents. It is not the ultimate aim of this study, but just another more explicit form of the SRS documents.

The railway domain is an environment where numerous heterogeneous information sources exist. The ERTMS system basically relies on information exchange. Ontologies provide a number of useful features for intelligent systems, as well as for knowledge representation generally. The ERTMS ontology that we propose also aims at offering a solution for information exchange, and this for a better railway transportation world.

Train control is an important part of any railway operation management system. In the past a number of different Automatic Train Control (ATC) systems have evolved in different countries at different times. Due to the incompatibility and lack of interoperability among these systems, as well as to a significant increase in density of train traffic anticipated, many railways rethink their infrastructure strategy, in order to accommodate high levels of traffic, in which ATC systems play an important part. This and the fact that many railways would like to introduce standardized systems to reduce system costs are, among others, the reasons of the existence of this system. In order to establish international standardization of ATC systems, the SRS document specifies the European Rail Traffic Management System/European Train Control System (ERTMS/ETCS).

The ERTMS System Requirements Specification is a set of documents written in natural language, English in this case. It specifies the European Rail Traffic Management System/

European Train Control System (ERTMS/ETCS) which is a control and signalisation innovative system of the railway vehicles and tracks. Also, system safety plays an important role in railway transport as it constitutes a challenging issue that has engaged strong and continuous research interest.

B. *Ontology building from normative documentation*

As mentioned before, in this ERTMS ontology, concepts are formalized as classes (terms). An ontology is not only the identification and classification of concepts, but also of their inherent characteristics that are here called ”properties”. Moreover the relations gather the concepts together. Primarily, we used the ”is-a” relation which is a subsumption relation allowing the formal heritage of properties. The ”has-a” relation, also known as composition, is used as well in this ontology, this time not for the class layer but for the instance layer. If, at the beginning, we had conceived our primary concept structure using the two relations for the classes, a differentiation became crucial as work proceeded. Then, other relations were established according to the system's syntax. These relations are created based on properties declaration and domain specification (tab allowing to select the class(es)) on which they take effect. Our ontology is structured into several modules. The Entity module, i.e. the superclass containing several entities like Driver, ERTMS, Procedure, describes entities that are used to define the required system behavior on a context level. The OSI_Model is a sibling class of Entity, a module aiming at describing the Open Systems Interconnection (OSI) model. This is a conceptual model that characterizes and standardizes the internal functions of a communication system by partitioning it into abstraction layers. This module will be more thoroughly explained in section VI. Another sibling class of the above mentioned ones is Source. It formalizes information about the SRS and other ERTMS/ETCS documents used as corpus of these ontology. TrainCategories is also a child of the Entity superclass, containing information about the different types of rolling stock.

V. Modelling ERTMS procedures

In figure 2, we present an example of a procedure defined in the SRS called ”Entering SH mode”. ”The ”Shunting” mode is, by definition, a type of ERTMS/ETCS on-board equipment allowing a train to move without having the update train data.

There exist several ERTMS operating modes, as well as all operational modes and procedures necessary to ensure safe information exchange between the driver and the embedded subsystem. Each mode is associated with a specific configuration (train, track and conditions) defining the system state.

Transitions between modes require the establishment of different conditions required to perform the transition properly, i.e. safely. In the SRS, the procedures associated or involving mode transistions are defined by flowcharts linking conditions, decisions and states. The ”shunting” flowchart is presented in figure 2.

In order to catch the semantics of these flowcharts in our ontology, we transformed the state transitions in each flowcharts into rules expressed in the SWRL language provided by the

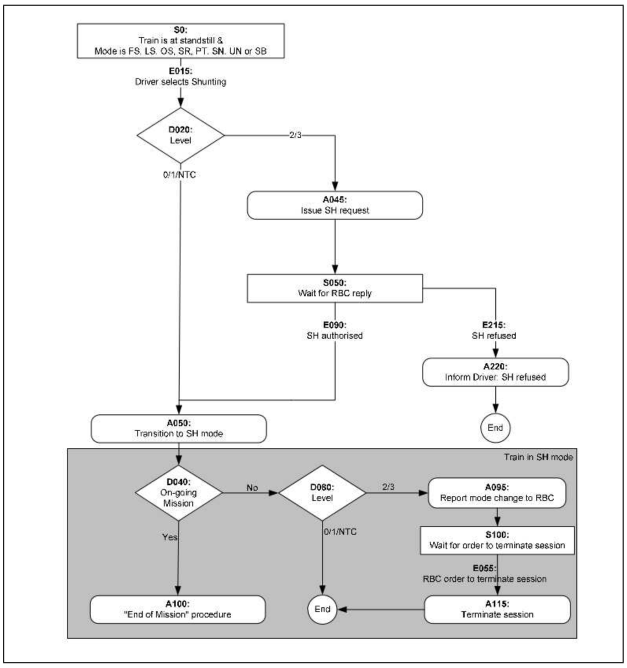

Fig. 2. Flowgraph for the ”Shunting” procedure

Protégé framework. SWRL stands for ”Semantic Web Rule Language” and provides a syntax and a semantics to express rules upon the entities available in the ontology. SWRL rules have the form of an implication between an antecedent (body) and consequent (head). The intended meaning can be read as: whenever the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold.

Considering the flowcharts as state-transition machine, we derived a flowchart into a set of SWRL rules, each rule corresponding to a transition. By doing this, we intend to catch the dynamic behavior of ERTMS/ETCS control system.

VI. Focus on the radio-communication part

As mentioned in the sections before, this ontology is constructed by modules. One of these modules formalises the OSI (Open Systems Interconnection) model and another sub-module deals with the application of the generic OSI model to the ERTMS system. This section presents the generic telecommunication model, followed by its instanciation with the OSI model and finaly with the ERTMS telecommunication subsystem.

A. *The radio telecommunication model*

Firstly, we defined a generic radio telecommunication model. This model/module is composed of several concepts[1]:

- the `NetworkStack` is the telecommunication stack which is composed of several `Layer` s.
- the `Layer` is a part of the `NetworkStack` which is able to marshall and unmarshall some Messages. Each `layer` is linked to two other `Layers`: an upper layer and a

[1]concepts defined in our ontology will be typesetted like this `ConceptName`

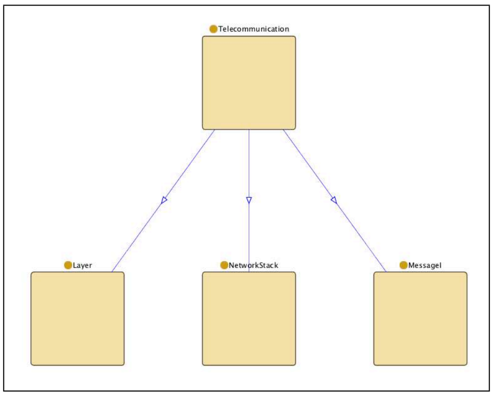

Fig. 3. Generic radio telecommunication concepts

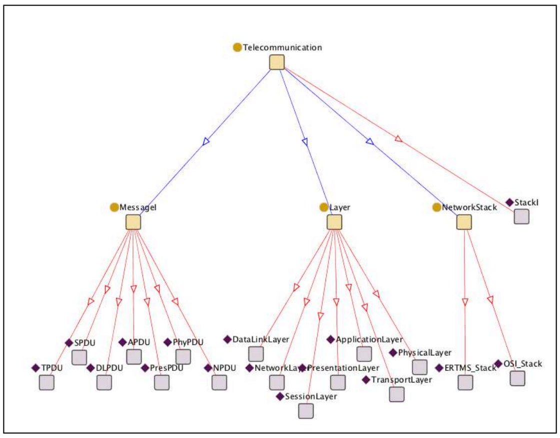

Fig. 4. Hierarchy and instances for the OSI model

down layer. The combination of this set of layers is a `NetworkStack`. A `layer` manipulates some `Messages`.

- the `Message` defines the data that will be sent and received on the network by the `Layers`.
- The `Telecommunication` concept references the concepts defined above.

Figure 3 presents graphically this set of concepts.

B. Feeding the ontology with the OSI model

Next, we populated the ontology with the concepts that describe the OSI model composed of 7 layers. This part of the work was usefull to see if the concepts defined into the radio telecommunication model were enough and to be sure that nothing was forgotten.

Figure 4 presents the hierarchy as defined previously and all the instances which represent the different layers of a classical OSI network stack. There is one important relation between the layers. That relation describes the link between two consecutive layers. Its is notated `hasUpperLayer` between layer N and N+1 and its opposite `hasDownLayer` between N and N-1. The uppest layer does not have an `hasUpperLayer`, nor does the lowest layer have a `hasDownLayer`. These relations are not shown on the figure 4 to keep a clear schema.

C. Feeding the ontology with the ERTMS radio subsystem

We applied the same reasoning to represent the concepts of the ERTMS/ETCS radio subsystem. This radio subsystem is

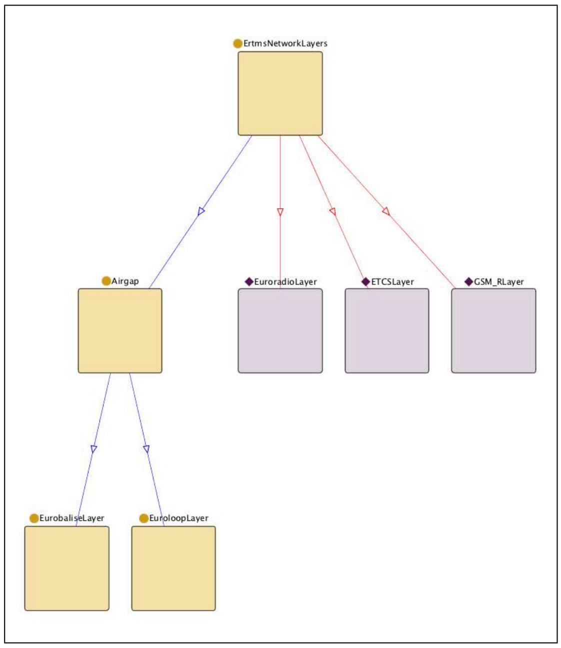

Fig. 5. Hierarchy and instances for the ERTMS radio subsystems

composed of three layers (from down layer to upper layer):

- the `GSM_RLayer` is based on the GSM specification with some modifications to fit the railway industry needs. The goal of this layer is to transport data packets through a celullar network between the train and the Radio Block Center (RBC).
- the `EuroradioLayer` deals with the end to end communication between an embedded application into the train and an application on ground. This layer is also responsible for non functional properties like authentication and crytptography of the messages.
- the `ETCSLayer` manages the messages at the application level of ETCS. This layer permits the communication between the onboard EVC and the ground system RBC that gives the movement authority grant(s) to the train.
- `AirGap`, `EurobaliseLayer` and `EuroloopLayer` represent equipements put on the track. These equipements communicate with the train when it goes over the equipments.

Figure 5 shows the ERTMS Network layer stack with three instances that correspond to the layers described just before.

D. Current state

This ERTMS ontology is structured into several layers. The `Thing` superclass contains several classes like `Entity`, `Source`, `OSIModel`, etc which, in their turn, have several subclasses. For example, the `Entity` class reunites the subclasses `Driver`, `ERTMS` and `Procedure`. The `ERTMS` subsubclass contains `ApplicationLevel`, `ERTMSNetworkLayer` and `ETCS`. These are just a few examples of terms that we enterred in the surface levels of the class structure of this ontology.

Currently, the ERTMS ontology that we have been creating contains 112 classes, 193 instances, and 104 properties including object, datatype and annotation properties.

VII. Related works

Due to the lack of space, we won't provide a huge panel of related works. With a few references, we will show that the main aspects of our work have already been studied and that there exist a solid background to tackle now with complex railway systems (like ERTMS/ETCS is) while involving several concerns like formalisation, requirements engineering, traceability, ...

A. Ontologies and software engineering

- In [7], an ontology called *OntoTest* is presented. This ontology is developed in order to promote organization, reuse and sharing of software testing knowledge. The main concepts and artefacts of testing are described (Process, phases, resources, procedures). The ontology itself is figured with UML class diagrams, W3C formalisms are not used in the paper but the ontology is now available in OWL format.
- The work presented in [8] is very close to the goals of our work. Starting from an industrial-use case (the Onboard Unit of ERTMS) a methodology to improve the testing process is provided. This methodology involves the analysis of the SRS specifications, the rewriting of the requirements into a "formal" language. The definition of this language is based on a previously established ontology classicaly defining the concepts, relations and axioms of the domain.

B. Ontologies and requirements engineering

[9] describes the expected benefits but also the challenges of using ontologies in requirements engineering (RE) activities. This is exactly the basis of our approach. The main statement is that such approach needs the definition of three ontologies : (1) an **application domain** ontology, (2) a **requirements** ontology and (3) a requirements specification **document** ontology. The **application domain** ontology calls itself a double-utility ontology i.e. a **domain** ontology that defines the necessary concepts for all training in the domain, and an **application** ontology which is one defining the concepts specific to a given application or méthod. The **requirements** ontology is used for representing requirements and their various relationships, as well as the relationships between requirements and systems. Whereas the requirements specification **document** ontology is a documentary ontology. The present paper deals with the creation of an ontology the first type presented above.

C. Ontologies and railway systems / applications

In [10], the authors present an ontology creation work conducted during the FP6 InteGRail project [11]. They used the same tools as us (OWL, Protege) to modelize an ontology that permits to check network statement for infrastructure operators. Using the ontology, they combine the network statements of different countries in different formats and analyse them in a transparent way. They modelized the network using concepts like network node, network line, track section, track node. All these concepts permit to the authors to represent the railway network as an object graph. In our work we could reuse such concepts.

TABLE I
Ontology pitfalls scanner OOPS!

	Pitfall	Cases
P04	Creating unconnected ontology elements	7
P05	Defining wrong inverse relationship	2
P08	Missing annotations	244
P11	Missing domain or range in properties	35
P13	Missing inverse relationships	33
P19	Swaping intersection and union	38
P21	Using a miscellaneous class	2
P22	Using different naming criteria in the ontology	ontology* (?)
P24	Using recursive definition	4

VIII. Conclusions and perspectives

In the present paper, we presented an experimental approach aiming at establishing an ontology of a complex domain like the ERTMS/ETCS railway control system. This development is mainly based on the study of a set of referential texts. As an example of the benefits we expect to obtain, we presented the enrichment of the ontology with the consideration of OSI standard levels to define precisely the concepts regarding the radio communication aspects of ERTMS.

SRS coverage: Since we focused on a first feasibility of the approach, the current coverage of the available texts by our ontology is obviously reduced. This work shall be improved in order to make our "ontological product" actually usable. It would be a painstaking work that could possibly take advantages on techniques (and related tools) such as automatic language processing. For example, the following step of our experiment may be the use of the GATE framework [12], since it provides ontological and also machine learning facilities.

Ontology quality: The quality of the ontology, viewed as a product, can be twofold: first, the quality of the embedded knowledge (as a semantic object); and second, the quality of the ontology itself (as a syntactic object).

The second point can be treated by the use of experience feedback from the elaboration of other ontologies and, particularly by taking into account the best practices of the domain sometimes identified and integrated into dedicated static analysis tools (like OOPS! [13]).

Table I gives the results of the evaluation of the ERTMS ontology by the tool OOPS!, in its current state. These results are barely correct because this "syntactic" aspect of the assessment has not been taken into account yet. For example, the pitfall with the worst score should be easily corrected simply by using the correct "annotation property" attribute for the definitions.

It is also possible to improve the overall quality (structure) of the current ontology by studying aspects like modularity ([14]), thus improving the decomposition and the potential reuse of the knowledge. A better modularity should also make easier the reuse of other related ontologies like testing ontologies, RE ontologies (as stated in section VII-B).

Usage of reasoners: The first point (knowledge quality) is essentially a matter of specific expertise to the considered field, but it can also be enhanced by the power of inference

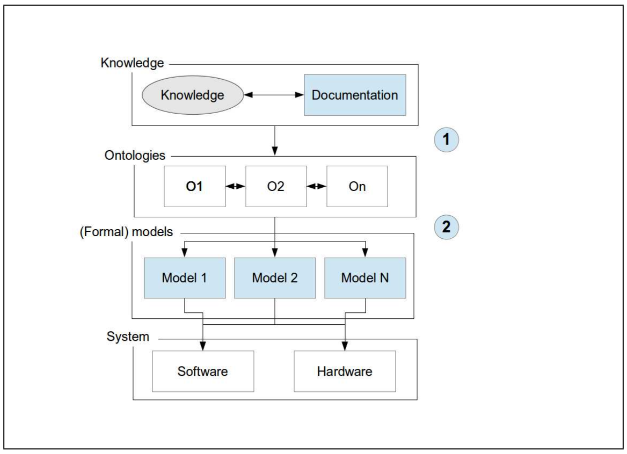

Fig. 6. Ontologies in the formalized developpement of safety critical systems.

mechanisms used especially to detect semantic inconsistencies, incompleteness of relations...

All these criteria are not assessable by previous techniques (syntactic/structure level). As we study complex specification documents, it is even more important to implement these mechanisms earlier in the development process, so as to achieve a real "debugging" of the ontology before effective implementation of the system.

Linking ontology and external (formal) models: As a next step, when the ERTMS ontology will be rich enough to be usable, we will start to tackle the problem of deriving more concrete models (mainly formal ones). As described in figure 6, the current work (number one circled) deals with the analysis of available documentation and expert knowledge to derive one ontology (and probably several others in the future) which can be taken as a first step for an abstract formalization.

Formal methods are highly recommended for the development of safety-critical (railway) systems (cf. CENELEC 50128 Norm [15]). The ERTMS/ETCS is a system of this kind and, thus, a formalizable domain.

Indeed, in the openETCS project [16], a european large project involving the main actors of railway research, ERTMS/ETCS (semi-)formal models will be delivered (as well as the corresponding tool-chains). More than ten formalisms/approaches are studied. They range from ADA (the robust programming language), UML and/or sysML, to formal methods like SCADE, B or eventB, Petri nets…

The next step (number two circled) will be the derivation of more concrete models using available and well-known (semi-)formalisms like those used in the openETCS initiative. We intend to show that an initial formalization derived from an ontological conceptualisation will be helpful to define the architecture and the main properties for derived formal models.

Since the ontological support languages (OWL, SWRL, …) used while elaborating our ontology are not too far from classical first order logic and set theory, one path to explore may be a model transformation from our ontology into formal specifications expressed within a "classical" formalism such as the B formal method [17].

Clearly, connecting our approach to the artefacts (formal models!) of the openETCS will be a real achievement.

A. Acknowledgements

The present research work is supported by the ICSIT (International Campus on Safety and Intermodality in Transportation) program and funded by the Nord Pas de Calais French Region and the ERDF (European Research and Development Funds). The authors gratefully acknowledge the support provided by these institutions.

References

[1] D. Bjørner, "Rôle of domain engineering in software development—why current requirements engineering is flawed!" in *Perspectives of Systems Informatics*, ser. Lecture Notes in Computer Science, A. Pnueli, I. Virbitskaite, and A. Voronkov, Eds. Springer Berlin Heidelberg, 2010, vol. 5947, pp. 2–34. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-11486-1_2

[2] A. S. Tanenbaum, *Computer Networks*. Prentice Hall, 1996.

[3] G. Thomas R., "A translation approach to portable ontology specifications," *Knowledge acquisition*, vol. 5, pp. 199–220, 1993.

[4] J. Charlet, B. Bachimont, and R. Troncy, "Ontologies pour le web sémantique," *Action spécifique*, vol. 32, pp. 43–63, 2003.

[5] N. F. Noy and D. L. Mcguinness, "Ontology development 101: A guide to creating your first ontology," Online, 2001. [Online]. Available: http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noy-mcguinness.html

[6] E. U. G. UNISIG, *System Requirements Specification (SRS) version 3.2.0*, E. R. Agency, Ed., 2012. [Online]. Available: http://www.era.europa.eu

[7] E. F. Barbosa, E. Y. Nakagawa, A. C. Riekstin, and J. Madonado, "Ontology-based Development of Testing Related Tools," 2008. [Online]. Available: http://www.labes.icmc.usp.br/moduloeducacional/publicacoes/SK06Ellen.pdf

[8] G. Bonifacio, P. Marmo, A. Orazzo, I. Petrone, L. Velardi, and A. Venticinque, "Improvement of processes and methods in testing activities for safety-critical embedded systems," in *Computer Safety, Reliability, and Security*, ser. Lecture Notes in Computer Science, F. Flammini, S. Bologna, and V. Vittorini, Eds. Springer Berlin Heidelberg, 2011, vol. 6894, pp. 369–382.

[9] V. Castaneda, L. Ballejos, M. L. Caliusco, and M. R. Galli, "The use of ontologies in requirements engineering," *Global journal of researches in engineering*, vol. 10, no. Issue 6, Nov. 2010. [Online]. Available: http://www.engineeringresearch.org/index.php/GJRE/article/view/76/71

[10] S. Verstichel, F. Ongenae, L. Loeve, F. Vermeulen, P. Dings, B. Dhoedt, T. Dhaene, and F. D. Turck, "Efficient data integration in the railway domain through an ontology-based methodology," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 4, pp. 617–643, 2011.

[11] InteGRail Consortium, "Integrail, intelligent integration of railway systems." InteGRail Consortium, 2009. [Online]. Available: http://www.integrail.info

[12] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, N. Aswani, I. Roberts, G. Gorrell, A. Funk, A. Roberts, D. Damljanovic, T. Heitz, M. A. Greenwood, H. Saggion, J. Petrak, Y. Li, and W. Peters, *Text Processing with GATE (Version 6)*, 2011. [Online]. Available: http://tinyurl.com/gatebook

[13] M. Poveda-Villalón, M. Suárez-Figueroa, and A. Gómez-Pérez, "Validating ontologies with oops!" in *Knowledge Engineering and Knowledge Management*, ser. Lecture Notes in Computer Science, A. Teije, J. Völker, S. Handschuh, H. Stuckenschmidt, M. d'Acquin, A. Nikolov, N. Aussenac-Gilles, and N. Hernandez, Eds. Springer Berlin Heidelberg, 2012, vol. 7603, pp. 267–281.

[14] C. Bezerra, F. Freitas, J. Euzenat, and A. Zimmermann, "ModOnto: A tool for modularizing ontologies," in *Proc. 3rd workshop on ontologies and their applications (Wonto)*, Salvador de Bahia, Brésil, Oct. 2008, p. No pagination., bezerra2008a INRIA-CNPq-OntoCompo; IST-NeOn. [Online]. Available: http://hal.inria.fr/hal-00793533

[15] CENELEC, "Railway applications - communications, signalling and processing systems - software for railway control and protection systems," 2011.

[16] openETCS, ITEA2 openETCS consortium, 2012. [Online]. Available: http://openetcs.org

[17] J.-R. Abrial, *The B Book - Assigning Programs to Meanings*. Cambridge University Press, Aug. 1996.