# An ontology for the ERTMS/ETCS

Olimpia Hoinaru, Christophe Gransart, Georges Mariano, Etienne Lemaire

# An ontology for the ERTMS/ETCS

Olimpia Hoinaru[1], Christophe Gransart, Georges Mariano, Étienne Lemaire

*Université Lille Nord de France, French Institute of Science and Technology for Transport, Development and Networks (IFSTTAR) Villeneuve d'Ascq, France*

**Abstract**

In the present paper we aim to describe the research and practical work of creating an ontology of the new European system of control and signalling ERTMS/ETCS. The object of our study is the System Requirements Specification set of documents, that is the corpus of our ontology. This one was processed using an NLP tool i.e. Protégé, in order to obtain a more explicit and easily usable form of these specifications. This processing was performed through a research and practice of system modelling and knowledge formalisation by using the OWL and SWRL languages. To be noted that this work is still in process, and that the part presented here constitutes the first steps in the creation of this ontology.

*Keywords*: Knowledge Engineering; Requirements Engineering; Ontologies; Natural Language Processing (NLP); European Rail Traffic Management System (ERTMS); European Train Control System (ETCS); Railway Systems.

**Résumé**

Le présent article a pour but de décrire les travaux de recherche et développement d'une ontologie du système européen de contrôle-commande et signalisation ERTMS/ETCS. L'objet de notre étude est l'ensemble des documents appelés « System Requirements Specifications » spécifiant le cahier des charges du système évoqué. Ces documents ont été traités à l'aide d'un outil de TALN (Traitement Automatique du Langage Naturel) à savoir Protégé, un outil de création d'ontologies. Ce traitement est fait par une recherche et application des domaines de la modélisation de systèmes et de la formalisation des connaissances, en utilisant les langages OWL et SWRL. À noter que ces travaux sont encore en cours de réalisation. La partie présentée dans cette contribution constitue une première étape de ce processus.

*Mots-clé*: Ingénierie des Connaissances; Ingénierie des Exigences; Ontologies; Traitement Automatique du Langage Naturel (TAL); European Rail Traffic Management System (ERTMS); European Train Control System (ETCS); Système ferroviaire.

[1] Corresponding author. Tel.: +33-320-43-8333; fax: +33-320-43-8359.
*E-mail address*: olimpia.hoinaru@ifsttar.fr

## 1. Introduction and context

The research and practical work that we performed for this article consisted in the processing of the available documents describing the System Requirements Specifications of the European Rail Traffic Management System (ERTMS). This contribution will tackle the process of creation of this ontology, as well as some information introducing ontologies in the knowledge engineering and requirements engineering.
Train control is an important part of any railway operations management system. In the past, a number of different Automatic Train Control (ATC) systems have evolved in different countries at different times. These systems are incompatible and not interoperable. Only a few of these systems are used in more than one country, and even in those cases there have been differences in detailed development which have resulted in incompatible and not interoperable versions.
Many railways anticipate a significant increase in density of train traffic and are rethinking their infrastructure strategy in order to accommodate high levels of traffic, in which ATC systems play an important part. Also many railways would like to introduce standardised systems to reduce system costs. The SRS set of documents specifies the European Rail Traffic Management System/European Train Control System (ERTMS/ETCS). Its aim is to establish international standardisation of ATC systems.
The modelling and formalisation of such a complex structure is still the aim of our research. The work presented in this paper only describes the first steps of this process. Compared to other types of approaches of processing the specification documents of the ERTMS/ETCS system, who favour formal methods (e.g. Petri nets, B method, etc), we preferred to create an ontology. In Figure 1 the general evolution of the technologies used for this matter is highlighted.
By doing this, we follow the paradigm stated in (Bjørner, 2010): "Before software can be designed we must understand the requirements. Before requirements can be finalised we must have understood the domain". But where Dines Bjørner uses pure formal logic to tackle generic sample problems, we will experiment the use of ontological technologies (conceptualisation, formalisation, reasoning) to tackle a real and complex system.

## 2. General goals

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

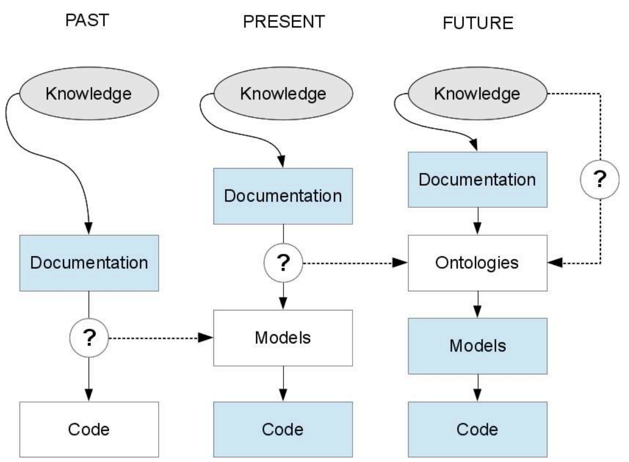


Fig. 1. Ontologies for software-based systems

The ERTMS ontology models and formalises the System Requirements Specification documents of the ERTMS that are written in natural language. The aim of this ontology is the formalisation 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) (Tanenbaum, 1996) model and another one concerns the application of the OSI model to the ERTMS/ETCS subsystem dealing with data transmission by means of radiocommunication.
We aim that, when in its final version, the ERTMS ontology represent a structured and comprehensible form of the ERTMS SRS. It should be able to provide help to human agents in a comprehension and reuse process as well as to programs. System modelling tools could take the owl or xml file obtained as a lexical ressource and use it for information extraction or system modelling purposes.

## 3. Elaborating ontologies

In this section, an account of the definition and usage of ontologies will be provided.

### *3.1. Defining ontologies*

Ontologies are formal representations of knowledge of a certain domain. Several definitions of the term “ontology" have been provided. Gruber poses that “an ontology is an explicit specification of a conceptualization". According to the same author, “the term is borrowed from philosophy, where an ontology is a systematic account of Existence" (Gruber, 1993a).
There are four types of information allowing us to be clear about what we represent in an ontology. These are the types 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 formalised 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 formalised as classes together with several types of properties and the relations among them. The so-called “rules" are created for the purpose of modelling requirements and certain “behaviours" 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.
Furthermore, we will present some of the ontology development methodologies existing and explain more thoroughly the methodology that we used in the framework of our project.

### *3.2. Methontology*

“Methontology" is the term used to describe one of these methodologies for creating an ontology. It is among the more comprehensive ontology engineering methodologies as its aim is to build 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 for creating ontologies. Other methodologies, such as the corpus-based methodology exist. [4] explains this methodology in a very pedagogical way. In this paragraph, we will briefly present this methodology and in a later section, we will explain how we applied it in the case of our ontology.

### *3.3. The corpus-based methodology*

The corpus-based methodology helps to acquire ontologies starting from corpora. This method was developed by B. Bachimont (Charlet et al., 2003). The four steps of this method are:

- Corpus analysis. This first step implies finding texts written in natural language that explain the knowledge of the stated domain, for purposes of practice, specification or knowledge transfer. The collation of these texts is called corpus collection and preparation. The preparation of the corpus consists in thorough analysis of the textual resources that will be the source of allowing us to characterise the useful concepts for modelling an ontology and the corresponding semantic content. This corpus approach implies terminology extraction tools, natural language processing tools but also language processing carried out manually for a better disambiguation.
- Semantic normalisation. This second step uses the candidate terms defined at the first stage, terms that have a meaning for the reader. But nothing makes sure that this sense is unique. On the contrary, we are in the case of a linguistic behaviour where meanings are ambiguous, definitions not always clear or well-formed, and

they depend particularly on the context of the user's interpretation. However, in the ontological modelling, we seek to build primitives whose meaning does not depend on other primitives and especially non-contextual ones. That is why, in order to obtain formal results, we need to normalise the meanings of the terms i.e. only retain one meaning for each term, one interpretation for human understanding. The corollary of the methodology, the meaning of a concept, is generally defined according to what it is (heritage relation to the father concept) and to what it is not (differences with the father or the sibling concepts). At the end of this step, we can assert that we have a "regional ontology".

- Ontologic engagement. This third step uses the regional ontology obtained at the step before with the aim of developing a formal ontology. This is the ontologic engagement. Formal semantics does not consider semantic notions anymore, but extensions i.e. the set of objects that satisfy properties defined in intension, at the preceding step. This formal ontology must take the form of a concept lattice.
- Ontology operationalisation. During this last stage, the approach is to represent the ontology in a knowledge representation language that allows inferential services. There exist several knowledge representation languages, such as the conceptual graphs and description logics. Both of them allow a certain number of operations on the ontologies: inferences proper to graph structures, classification into tree structures for description logics. The latter mentioned are the type of language used by the Web Semantics for representing ontologies and making inferences.

When using the Protege tool, as in our case, the OWL is the W3C standard used to develop the ontology.
(Noy et al., 2001) 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.

## 4. State of the art of ontology usage in Systems Engineering, Requirements Engineering and Railway systems

This section provides a panel of works related to the usage of ontologies in the domains of Systems Engineering, Requirements Engineering and Railway Systems. The aim of this section is to provide an overview of the main aspects of our work, as well as to state the fact that they have already been studied. Thus, there exists a solid background to tackle now with complex railway systems (like ERTMS/ETCS is) while involving several concerns like formalisation, requirements engineering, traceability, …

### *4.1. Ontologies and software engineering*

- In (Barbosa et al., 2008) an ontology called OntoTest is presented. This ontology is developed in order to promote organisation, reuse and sharing of software testing knowledge. The main concepts and artefacts of testing are described (processes, 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 (Bonifacio et al., 2011) 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 classically defining the concepts, relations and axioms of the domain.

### *4.2. Ontologies and requirements engineering*

Castaneda et. al. (2011) 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 an 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 is 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 method. 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 of the first type presented above.

### *4.3. Ontologies and railway systems/applications*

In (Verstichel et al., 2011), the authors present an ontology creation work conducted during the FP6 InteGRail project (InteGRail). They used the same tools as us (OWL, Protégé) to model an ontology that permits them to check network statement for infrastructure operators. By using the ontology, they combine the network statements of different countries in different formats and analyse them in a transparent way. They modelled the network using concepts such as network node, network line, track section, track node. All these concepts allow the authors to represent the railway network as an object graph. In our work we could reuse such concepts.

## 5. Global view of the proposed ERTMS ontology

This section presents the normative documentation and the methodology that we followed for acquiring this first version of our ERTMS ontology.

### *5.1. The ERTMS/ETCS System Requirements Specification – the normative documentation*

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.

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, 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.

### *5.2. The chosen method*

The method that we chose for the creation of this ERTMS ontology is the corpus-based methodology that is detailed and explained in section 3 (Charlet et al., 2003). This subsection presents our approach of constructing our ERTMS ontology, based on studying normative documentation.

For the first stage of this methodology i.e. **corpus analysis** we chose our corpus among the documents that exist in the domain of railway transport. One of these documents is the System Requirements Specification document that was imposed in the framework of this project, as it is essential for the system. It is actually a set of documents (SRS 3.2.0) provided by the European Railway Agency (ERA). Other related documents are the “ERTMS Glossary" (ERTMS Glossary, 2012) and the “ETCS Implementation Handbook" (ETCS, 2008) published by the 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 and comprehension of these documents, and on the transposition of the information conceptualised in the same documents. All this was carried out manually by 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 human understanding. So, in the case of this first part of our ERTMS ontology, for the preparation of the corpus, only language processing carried out manually was used, for the reasons mentioned below. For the second step i. e. the **semantic normalisation**, the candidate terms identified at the first step were entered in an OWL format document using the Protégé tool. We began thus to build our ontology from normative documentation.

During the third step of the methodology i.e. the **ontologic engagement**, the concepts of the domain were formalised 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 also used in this ontology, this time not for the class layer but for the instance layer. Since, 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 were created based on property declaration and domain specification (tab allowing us to select the class(es)) on which they take effect.
For the fourth step i.e. the **ontology operationalisation** of the creation of this part of our ontology, we used description logics that are available in the Protégé tool. 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 behaviour 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 characterises and standardises the internal functions of a communication system by partitioning it into abstraction layers. This module will be more thoroughly explained in section 6.
- another sibling class of the above-mentioned one is `Source`. It formalises 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.

After the representation of the concepts of a domain, the next step is to make inferences. For the moment, our ERTMS ontology does not contain this type of processing, but it is intended for future actions.

## 6. 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. This section presents the generic telecommunication model, followed by its instantiation with the OSI model and finally with the ERTMS telecommunication subsystem.

### *6.1. The radio telecommunication model*

First, we defined a generic radio telecommunication model (Figure 2). This model/module is composed of several concepts:

- the `NetworkStack` is the telecommunication stack which is composed of several `Layers`
- 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`: a higher `Layer` and a lower `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.

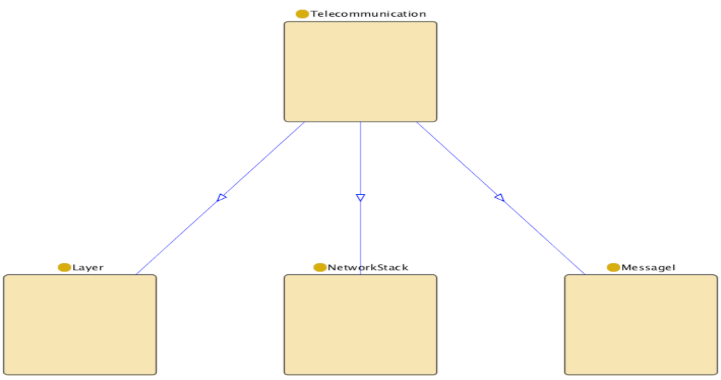


Fig. 2. Generic radio telecommunication concepts

### *6.2. 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 useful to see if the concepts defined into the radio telecommunication model were enough and to be sure that nothing was forgotten.

Figure 3a 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. The relation between layers N and N+1 is `hasHigherLayer`, and its opposite `hasLowLayer` between N and N-1. The highest layer does not have a hasHigherLayer, nor does the lowest layer have a `hasLowLayer`. These relations are not shown on the Figure 3a in order to keep a clear schema.

*6.3. Feeding the ontology with the ERTMS radio sub-system*

We applied the same reasoning to represent the concepts of the ERTMS/ETCS radio sub-system. This radio sub-system is composed of three layers (from lower layer to higher layer):

- `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 cryptography of the messages.
- the `ETCSLayer` manages the messages at the application level of ETCS. This layer permits the communication between the on-board EVC and the ground system RBC that gives the movement authority grant(s) to the train.
- `AirGap`, `EurobaliseLayer` and `EuroloopLayer` represent equipments put on the track. These equipments communicate with the train when it goes over them.
- Figure 3b shows the `ERTMS Network` layer stack with three instances that correspond to the layers described just before.

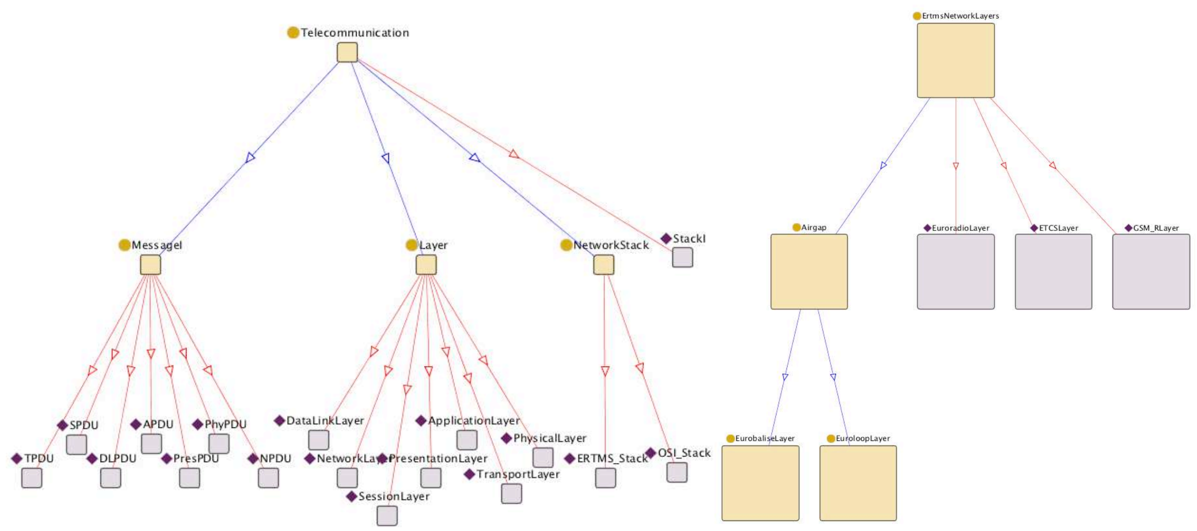


Fig. 3a. Hierarchy and instances for the OSI model. 3b. Hierarchy and instances for the ERTMS radio subsystems

**7. Results and discussion**

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

### *7.1. Results*

Currently, the ERTMS ontology that we have been creating contains 112 classes, 193 instances, and 104 properties including object, datatype and annotation properties. The maximum depth level is 9.

### *7.2. 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 (Cunningham et al., 2011), since it provides ontological and also machine learning facilities.

### *7.3. Modelling ERTMS/ETCS procedures*

We have modelled the ERTMS procedures defined in the SRS. For instance, the "Shunting" mode is, by definition, a type of ERTMS/ETCS on-board equipment allowing a train to move without having the updated train data. Considering the flowcharts as state-transition machines, we transcribed a flowchart into a set of SWRL rules, each rule corresponding to a transition. By doing this, we intended to catch the dynamic behaviour of the ERTMS/ETCS.

### *7.4. 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!) (Poveda et al., 2012).
Table 1 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 (Bezerra et al., 2008), 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.

### *7.5. Usage of reasoners*

Knowledge quality is essentially a matter of specific expertise to the considered field, but it can also be enhanced by the power of inference 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.

### *7.6. 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 4, 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 formalisation.
Formal methods are highly recommended for the development of safety-critical (railway) systems (cf. CENELEC 50128 (EN50128, 2011). The ERTMS/ETCS is a system of this kind and, thus, a formalizable domain.
Indeed, in the openETCS project (openETCS, 2012), 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 formalisation 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, etc) 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 (Abrial, 1996).
Connecting our approach to the artefacts (formal models) of the openETCS could be a perspective of this work.

| Pitfall number | Pitfall description | 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 | Swapping intersection and union | 38 |
| P21 | Using a miscellaneous class | 2 |
| P22 | Using different naming criteria in the ontology | ontology* (?) |
| P24 | Using recursive definition | 4 |

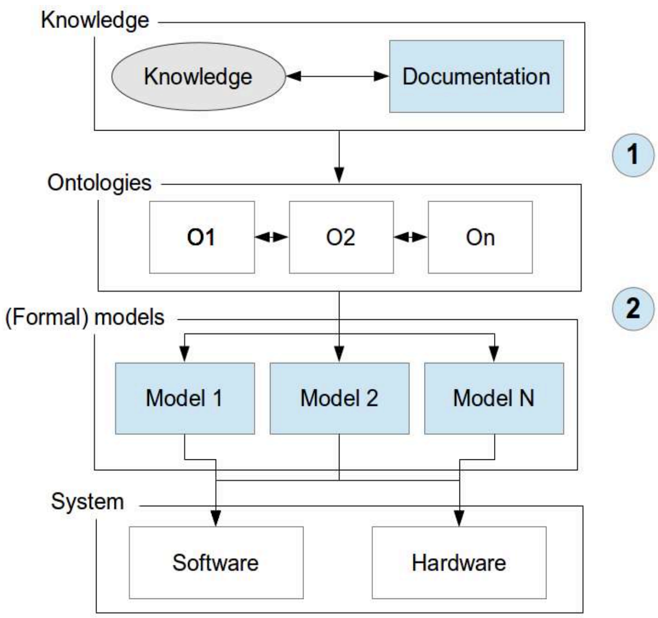


Fig. 4. Ontologies in the formalised development of safety critical systems

## 8. 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.

**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**

Abrial J.R. (1996). *The B-Book: Assigning Programs to Meanings*, Cambridge University Press, ISBN 0-521-49619-5.

Barbosa E. F., Nakagawa E. Y., Riekstin A. C., Madonado J. (2008). Ontology-based Development of Testing Related Tools. University of Sao Paulo – ICMC/USP

Bezerra C., Freitas F., Euzenat J., Zimmermann A.. (2008) ModOnto: A tool for modularizing ontologies. In *Proc. 3rd workshop on ontologies and their applications (Wonto)*.

Bjørner D. (2010) Rôle of Domain Engineering in Software Development—Why Current Requirements Engineering Is Flawed! In *Perspectives of Systems Informatics*, Pnueli, Amir & Virbitskaite, Irina & Voronkov, Andrei, Springer Berlin Heidelberg, pp. 2-34.

Bonifacio, G., Marmo, P., Orazzo, A., Petrone, I., Velardi, L., Venticinque, A. (2011). Improvement of Processes and Methods in Testing Activities for Safety-Critical Embedded Systems. In *Computer Safety, Reliability, and Security*, Lecture Notes in Computer Science, 6894, 369-382.

Castaneda V., Ballejos L., Caliusco M. L., Galli M. R. (2010). The use of ontologies in requirements engineering. In *Global journal of researches in engineering*, 10, 6.

Charlet J., Bachimont B., Troncy R. (2003). Ontologies pour le Web sémantique. In *Action spécifique*; 32, 43-63.

EN 50128. (2011). CENELEC,"Railway applications-communications,signalling and processing systems - software for railway control and protection systems"

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

Gruber, Th. R. (1993a). A translation approach to portable ontology specifications. In *Knowledge acquisition*, 5, 199-220

Gruber Th. R. (1993b). Towards Principles for the design of ontologies Used for Knowledge Sharing. In *Proceedings*. N. Guarino et R. Poli, *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Deventer, The Netherlands, Kluwer Academic Publishers.

Noy N.F., Mcguinness D.L. (2001). Ontology Development 101: A Guide to Creating Your First Ontology.

openETCS (2012). ITEA2 openETCS consortium, *http://openetcs.org.*

Poveda-Villalón, M., Suárez-Figueroa M.C., Gómez-Pérez A., (2012) Validating Ontologies with OOPS!. In *Knowledge Engineering and Knowledge Management*, Lecture Notes in Computer Science, Teije, A., Völker, J., Handschuh, S., Stuckenschmidt, H., d'Acquin, M., Nikolov, A., Aussenac-Gilles, N., Hernandez, N., Springer Berlin Heidelberg. 7603, 267-281.

Tanenbaum, A. S. (1996). *Computer Networks*. Prentice Hall

Verstichel S., Ongenae F., Loeve, Leanneke, Vermeulen, Frederik, Dings, Pieter, Dhoedt, Bart, Dhaene, Tom, Turck, Filip D. (2011). Efficient data integration in the railway domain through an ontology-based methodology. In *Transportation Research Part C: Emerging Technologies*. 19, 617-643.

UNISIG, ERTMS Users Group (2012). *System Requirements Specification* (SRS) version 3.2.0, European Raiway Agency.