CRYSTAL

CRitical S**YST**em Engineering **A**cce**L**eration

State of the art for RAIL ontology

D504.010

DOCUMENT INFORMATION

Project	CRYSTAL
Grant Agreement No.	ARTEMIS-2012-1-332830
Deliverable Title	State of the art for RAIL ontology
Deliverable No.	D504.010
Dissemination Level	RE
Nature	R
Document Version	V1-0
Date	2014-03-31
Contact	Renato De Guglielmo
Organization	Ansaldo STS (ASTS)
Phone	+39 (0)81 243 7608
E-Mail	Renato.DeGugliemo@ansaldo-sts.com

Authors

Name	Company	E-Mail
Renato De Guglielmo	Ansaldo STS	Renato.DeGugliemo@ansaldo-sts.com
Roberto Nardone	Ansaldo STS	Roberto.Nardone.Prof608@ansaldo-sts.com
Gianpaolo Massaroli	Ansaldo STS	Gianpaolo.Massaroli.Prof644@ansaldo-sts.com

Reviewers

Name	Company	E-Mail
Luigi Velardi	Ansaldo STS	Luigi.Velardi@ansaldo-sts.com
Pascal Poisson	Alstom Transport	pascal.poisson@transport.alstom.com
Christoph Scherrer	Thales Austria GmbH	christoph.scherrer@thalesgroup.com

Change History

Version	Date	Reason for Change	Pages Affected
0-1	07/03/2014	Create document	All
0-2	18/03/2014	Introduction of internal review comments and updates	All
1-0	31/03/2014	Introduction of external review comments and updates	All

CONTENT

Content of Figures

Content of Tables

1 Introduction

1.1 Role Of Deliverable

In a project such as CRYSTAL, which involves a lot of companies (within and outside the rail domain) it's needed to optimize communication efforts among partners and improve the quality of the work packages results.

The purpose of this deliverable is to collect the existing standards and the ontology catalogues related to the rail domain and to evaluate them regarding their relevance and acceptance, and then, to discuss and summarize ontology gaps or contradicting catalogues. This will constitute the basis for the following deliverables (especially for D504.021 and D504.022 –“Ontology definition”, V1 and V2-), where a railway ontology catalogue will be defined based on the here presented results.

Exploring the existing ontology catalogues to create a solid basis for a widely accepted vocabulary will help the companies involved in this project to hit the above mentioned target because, through the use of a ontology catalogue, all the essential terms used in various deliverables and reports will be unified in order to increase readability and thus the quality of them (especially, a corresponding unified glossary could be automatically appended to every deliverable), and the communication among partners will be simplified.

Finally, a clear ontology is not only helpful for communication among partners, but it is also a pre-requisite for defining IOS (as underlined in the following section), and it's hence evident that building a domain ontology assumes a central role within this project.

1.2 Relationship With Other CRYSTAL Documents

As said above, there is a straight connection between this document and the deliverables D504.021 (“Ontology definition -V1”) and D504.022 (“Ontology definition -V2”), where a railway ontology catalogue based on the findings and the evaluation of this document will be described. However, it's important to underline that also all the other deliverables in the railway domain will be influenced by the findings on this ontology.

Then, due to the fact that each of CRYSTAL subprojects includes its own domain ontology work package (respectively, 2.9, 3.8 and 4.7 for the aerospace, the automotive and the health care domain), which provides the analogous deliverable on the state of the art for the respective domain ontology, there will be the opportunity to evaluate all these documents in order to identify all the potential commonalities and the standardization issues.

Finally, a strong link with the Subproject 6 and its deliverables is envisioned. Indeed, the capabilities of the currently developed CRYSTAL Interoperability Specification platform should fulfil the different types of needs expressed by the use case drivers, and the domain ontology should be a resource usable by the RTP platform services to perform this task.

1.3 Relationship With Other Projects

In order to collect the existing standards and the ontology catalogues related to the rail domain, the outcomes of several previous projects will be taken into consideration. Indeed, this document will include results of projects such as InteGRail and CESAR.

1.4 Structure Of This Document

This document is structured as follows:

- Section 1 (this Section) introduces the contents and the structure of the document, clarifying also the relationships with other documents related to CRYSTAL project and with results coming from other projects;
- Section 2 provides a general understanding of what an ontology is and fixes the most basic concepts related to this topic;
- Section 3 describes and summarizes the existing ontology catalogues related to the rail domain, and evaluates the related technologies, tools and standards;
- Section 4 provides a first evaluation of our findings, to see whether there are ontology gaps or contradicting catalogues, in order to understand if all important aspects of the rail domain are covered;
- Section 5 reports the list of acronyms used in this document;
- Section 6 reports the list of references.

2 Ontology Generalities

2.1 Goals And Motivations

Before going into more details about all the possible definitions of the term “Ontology”, it’s important to clarify the reason why the research on ontology has become increasingly widespread in the computer science community, and its importance has been recognized in a multiplicity of research fields and application areas, including knowledge engineering, database design and integration, information retrieval and extraction.

The answer has to be found in the fact that nowadays it's needed to optimize the communication efforts because there are too many subtle distinctions in terminology and meaning, even inside the same industrial domain.

Indeed, according to [Musen, 1992; Gruber, 1993], it’s needed <<to share a common understanding of the structure of information among people or software agents>>, and a common alphabet is not enough: <<XML is only the first step to ensuring that computers can communicate freely. XML is an alphabet for computers and, as everyone who travels in Europe knows, knowing the alphabet doesn’t mean you can speak Italian or French>> [Business Week, 2002]. Standard glossaries can help, but...

• Defining standard vocabularies is difficult and time-consuming.

• Once defined, standards don’t adapt well.

• Heterogeneous domains need a broad-coverage vocabulary.

• People don’t implement standards correctly anyway.

• Vocabulary definitions are often ambiguous or circular.

Ontology, hence, as it will be clear in the following, represents a solution that seems to be the most consistent in solving the interoperability and semantic heterogeneity problems for establishing an efficient communication baseline and for integrating heterogeneous resources.

2.2 Definition

The term “Ontology” derives from Ancient Greek ὤν (‘on’, present participle of εἰμί -‘eimi’, “To be”-, “Being”, “Existing”, “What exists”) + λόγος (‘logos’, “Word”, “Speech”, “Rational account”, “Knowledge”) which stands for the study of Being. Indeed, philosophical ontology is the most fundamental branch of metaphysics, and it’s concerned with the study of what is, of the kinds and structures of objects, properties, events, processes, and relations in every area of reality: ontology studies Being or Existence as well as the basic categories thereof, trying to find out what entities and what types of entities exist.

However, the term "in se” was coined only in the Late Scholasticism of the early 17th century, but this philosophical sense of the term is what Jacob Lorhard had in mind when he used for the first time the term “Ontology” around 1613, and this is also why Nathan Bailey's 1721 Oxford English Dictionary defined ontology (popularized as a philosophical term by German philosopher Christian Wolff) as "An Account of being in the Abstract" [ORG, 2007].

Even though the word ontology, as said, is taken from Philosophy, during the '90s, this term became relevant for the Knowledge Engineering community. From this perspective, in the last two decades many definitions about what an ontology is have been proposed in the literature.

One of the first definitions of the term was given by Neches [Neches et al., 1991], which explained what to do to build an ontology giving some quite vague guidelines. He defined the term as follows:

<<An ontology defines the basic terms and relations comprising the vocabulary of a topic area, as well as the rules for combining terms and relations to define extensions to the vocabulary>>.

Since then, several scientists have tried to formulate a well-accepted definition of the term, which was defined by the Standard Upper Ontology Working Group (SUO WG, an IEEE group with the aim of developing a standard for specifying an ontology to support computer applications such as data interoperability, information search and retrieval, and natural language processing) as follows [SUO WG, 2003]:

<<An ontology is similar to a dictionary or glossary, but with greater detail and structure that enables computers to process its content. An ontology consists of a set of concepts, axioms, and relationships which describe a domain of interest>>.

Among all the contributions for a well-accepted definition of the term, the most significant ones were those offered by Thomas Robert Gruber (an American computer scientist which did a foundational work in ontology engineering, becoming well known for his definition of ontologies in the context of Artificial Intelligence) and Nicola Guarino (an Italian computer scientist and researcher in the area of Formal Ontology for Information Systems), both of whom will be presented in the following Section.

2.2.1 From Gruber To Guarino

A couple of years after the apparition of the first engineering definition of the term, Gruber [Gruber, 1993 a] defined an ontology as follows:

<<An ontology is an explicit specification of a conceptualization>>.

This definition became the most quoted in literature and by the ontology community: indeed, based on Gruber's one, many definitions of what an ontology is were hereinafter proposed [Corcho, 2004].

Borst [Borst, 1997] modified slightly Gruber's definition as follows:

<<Ontologies are defined as a formal specification of a shared conceptualization>>.

Then, Gruber's and Borst's definitions have been merged and explained by Studer [Studer et al., 1998] as follows:

<<An ontology is a formal and explicit specification of a shared conceptualization. “Conceptualization” refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. “Explicit” means that the type of concepts used, and the constraints on their use are explicitly defined. “Formal” refers to the fact that the ontology should be machine-readable. “Shared” reflects the notion that an ontology captures consensual knowledge, that is, it is not private of some individual, but accepted by a group>>.

In the meantime, in 1995, Guarino and Giaretta [Guarino and Giaretta, 1995] opened up a new perspective, proposing to consider an ontology as:

<<A logical theory which gives an explicit, partial account of a conceptualization>>,

where a conceptualization was basically the idea of the world that a person or a group of people could have [Corcho, 2004].

It’s possibile to talk about a new perspective because Guarino and Giaretta's definition was further refined in [Guarino, 1998], where the Italian scientist tried to clarify – with respect to the past works – and reformulate the notions of ontology and conceptualization as it follows.

Firstly, Guarino contemplates the distinction between “Ontology” (with the capital ‘o’, as in the statement “Ontology is a fascinating discipline”) and “ontology” (with the lowercase ‘o’, as in the expression “Aristotle’s ontology”).

He underlines that the same term has an uncountable reading in the former case, and a countable reading in the latter, and that, if, on the one hand, the former reading is reasonably clear (as referring to a particular philosophical discipline), on the other hand, two different senses are assumed for the latter term by the philosophical community and the Artificial Intelligence community (and, in general, the whole computer science community).

Indeed, in the philosophical sense, it’s possible to refer to an ontology as a particular system of categories accounting for certain vision of the world; as such, this system does not depend on a particular language (e. g.: Aristotle’s ontology is always the same, independently of the language used to describe it). On the contrary, in its most prevalent use in AI, <u>an ontology refers to an engineering artifact, constituted by a specific vocabulary used to describe a certain reality, plus a set of explicit assumptions regarding the intended meaning of the vocabulary words</u>. This set of assumptions has usually the form of a first-order logical theory, where vocabulary words appear as unary or binary predicate names, respectively called concepts and relations (in the simplest case, an ontology describes a hierarchy of concepts related by subsumption relationships; in more sophisticated cases, suitable axioms are added in order to express other relationships between concepts and to constrain their intended interpretation).

Guarino chooses to adopt the AI reading, using the word “Conceptualization” to refer to the philosophical one (in this way, two ontologies can be different in the vocabulary used -using English or Italian words, for instance- while sharing the same conceptualization).

At this point, Guarino notes that the above introduced notion of conceptualization would require a suitable formalization, since it may generate some confusions. Indeed, he reminds that a definition of conceptualization has been given in [Genesereth and Nilsson, 1987] as a structure <D, **R**>, where D is a domain and **R** is a set of relevant relations on D. However, according to Guarino, the problem with Genesereth and Nilsson's notion is that it refers to ordinary mathematical relations on D (*extensional* relations), which reflect a *particular* state of affairs (e. g.: in the blocks world, they may reflect a particular arrangement of blocks on the table), whereas it would be needed to focus on the *meaning* of these relations, independently of a state of affairs, speaking hence of *intensional* relations, named as *conceptual relations*, reserving the simple term "relation" to ordinary mathematical relations.

From another point of view, it's possible to say that, by means of a conceptualization, humans isolate relevant invariances from physical reality (i. e.: they ascribe properties and ordinary relations to things - Figure 2-1-) on the basis of:

• Perception

• Cognition and cultural experience

• Language

This is possible because humans are able to ascribe properties and ordinary relations to things by means of concepts: a concept is the part of meaning corresponding to general principles, rules to be used to determine reference -typically, abstractions from experience-; the extension of a concept is an object, which represents the part of meaning corresponding to the concrete reference. Concepts describing relations (friend-of, father-of...) are called conceptual relations.

According to Guarino, while ordinary relations are defined on a certain domain, conceptual relations have to be defined on a *domain space*: he defines a domain space as a structure <D, W>, where D is a domain and W is a set of maximal states of affairs of such domain, also called *possible worlds* (e. g.: D may be a set of blocks on a table and W can be the set of all the possible spatial arrangements of these blocks).

Hence, a conceptualization for D can be now defined as an ordered triple **C** = <D, W, $\mathfrak{R}$>, where $\mathfrak{R}$ is a set of conceptual relations on the domain space <D, W>. Therefore, a conceptualization could be seen as a set of conceptual relations defined on a domain space (Figure 2-1).

At this point, Guarino contemplates the structure <D, **R**> introduced in [Genesereth and Nilsson, 1987]: since it refers to a particular world (or state of affairs), he calls it a *world structure*, and, due to the fact that a conceptualization contains many of such world structures -one for each world- they shall be called the *intended world structures* according to such conceptualization.

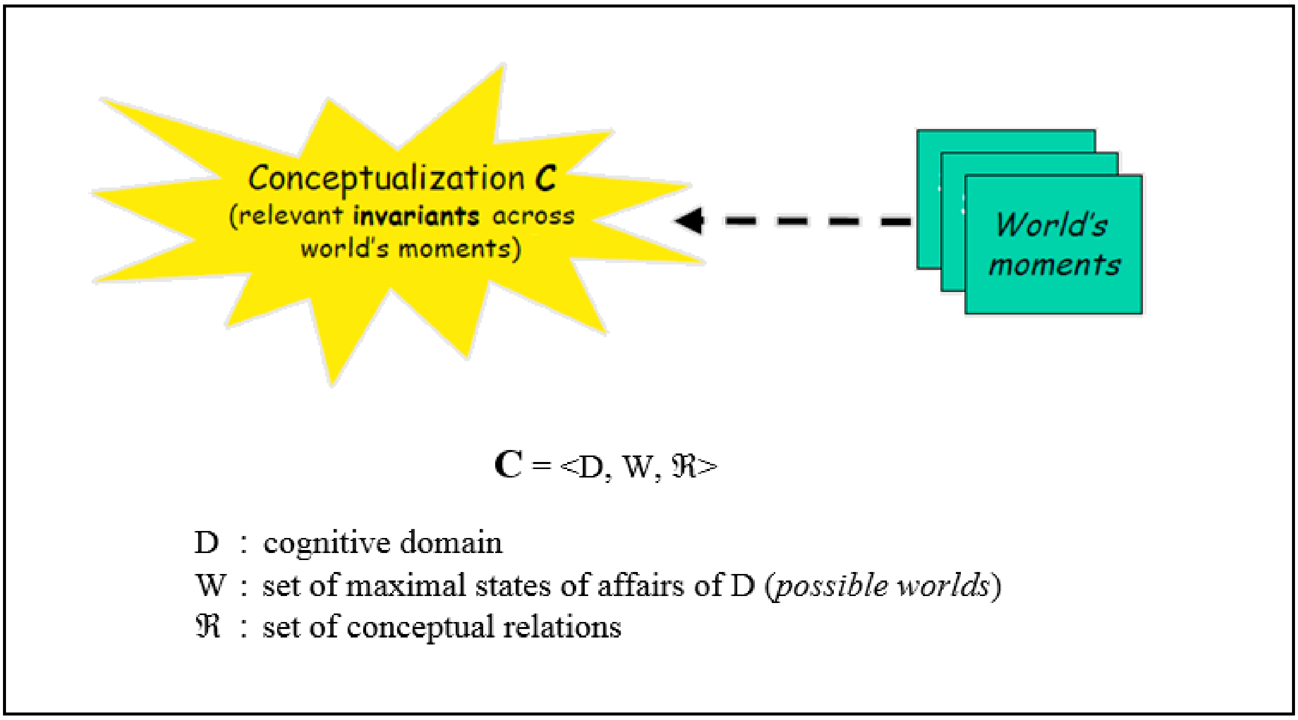

Figure 2-1: From experience to conceptualization

Then, by considering a logical language **L**, with a vocabulary V, he defines a *model* for **L** as a structure <**S**, I>, where **S** = <D, **R**> is a world structure and I: $V \rightarrow D \cup \mathbf{R}$ is an interpretation function assigning elements of D to constant symbols of V, and elements of **R** to predicate symbols of V. A model fixes therefore a particular *extensional* interpretation of the language.

Analogously, it's possible to fix an *intensional* interpretation by means of a structure <**C**, $\Im$>, where **C** = <D, W, $\Re$> is a conceptualization and $\Im$: $V \rightarrow D \cup \Re$ is a function assigning elements of D to constant symbols of V, and elements of $\Re$ to predicate symbols of V. Guarino names this structure as an *ontological commitment* (**K** = <**C**, $\Im$>) for **L**. The set $\mathbf{I}_K(\mathbf{L})$ of all models of **L** that are compatible with **K** will be called the set of *intended models* of **L** according to **K**.

Finally, the Italian scientist can clarify the role of an ontology, considered as a set of logical axioms designed to account for the intended meaning of a vocabulary: given a language **L** with ontological commitment **K**, an ontology for **L** is a set of axioms designed in a way such that the set of its models approximates as best as possible the set of intended models of **L** according to **K**. Hence, it's possible to say that an ontology **O** for a language **L** *approximates* a conceptualization **C** if there exists an ontological commitment **K** = <**C**, $\Im$> such that the intended models of **L** according to **K** are included in the models of **O** ($\mathbf{O}_K$), as depicted in Figure 2-2.

In this figure it's represented how the intended models of a logical language reflect its commitment to a conceptualization; an ontology indirectly reflects this commitment (and the underlying conceptualization) by approximating this set of intended models.

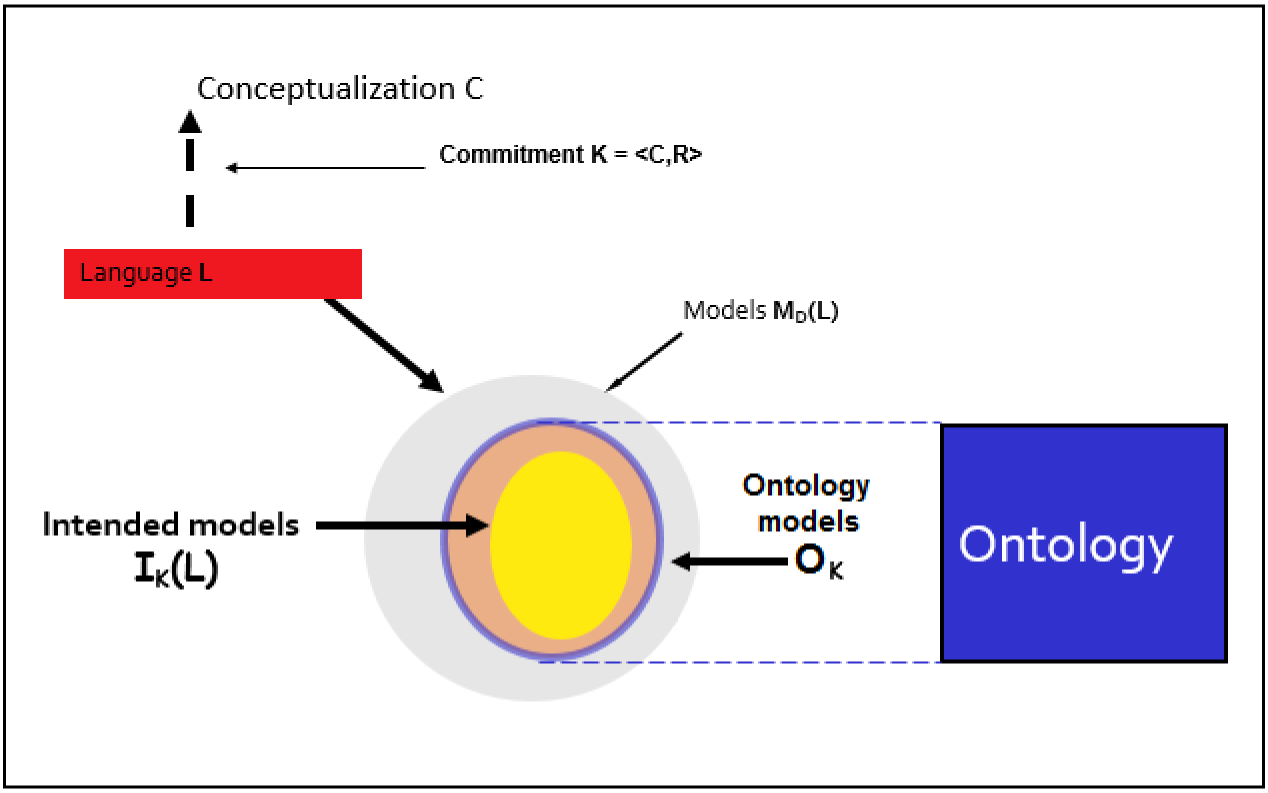

Figure 2-2: From conceptualization to ontology

With these clarifications, Guarino comes up to the following definition, which refines Gruber's one by making clear the difference between an ontology and a conceptualization:

<<An ontology is a logical theory accounting for the *intended meaning* of a formal vocabulary, i. e. its *ontological commitment* to a particular *conceptualization* of the world. The intended models of a logical language using such a vocabulary are constrained by its ontological commitment. An ontology indirectly reflects this commitment (and the underlying conceptualization) by approximating these intended models>>.

2.3 Characteristics And Classification

According to [Guarino, 1998], due to the fact that an ontology only indirectly accounts for a conceptualization, it's possible to classify ontologies depending on their *accuracy* in characterizing the conceptualization they commit to.

For an ontology there are three possible ways to get closer to a conceptualization:

- by developing a richer axiomatization;
- by adopting a richer domain and/or a richer set of relevant conceptual relations;
- by adopting a modal logic, which allows to express constraints across worlds, and by reifying worlds as ordinary objects of the domain.

Of course, there will be a tradeoff between a coarse and a fine-grained ontology committing to the same conceptualization: the latter gets closer to specify the intended meaning of a vocabulary (and therefore may be used to *establish consensus* about sharing that vocabulary or a knowledge base which uses that vocabulary), but it may be hard to develop and to reason on, both because the number of axioms and the expressiveness of the language adopted.

A coarse ontology, on the other hand, may consist of a minimal set of axioms written in a language of minimal expressivity, to support only a limited set of specific services, intended to be shared among users which *already agree* on the underlying conceptualization.

Speaking in terms of the earlier introduced formalism, we would say that, as a result of the ontology axioms, the set $\mathbf{O}_K$ has to properly cover $\mathbf{I}_K$. In general, however, there are five possible situations [Staab, 2004]:

1. $\mathbf{I}_K \cap \mathbf{O}_K = \emptyset$
2. $\mathbf{I}_K = \mathbf{O}_K$
3. $\mathbf{I}_K$ and $\mathbf{O}_K$ do properly overlap
4. $\mathbf{I}_K \subset \mathbf{O}_K$
5. $\mathbf{O}_K \subset \mathbf{I}_K$

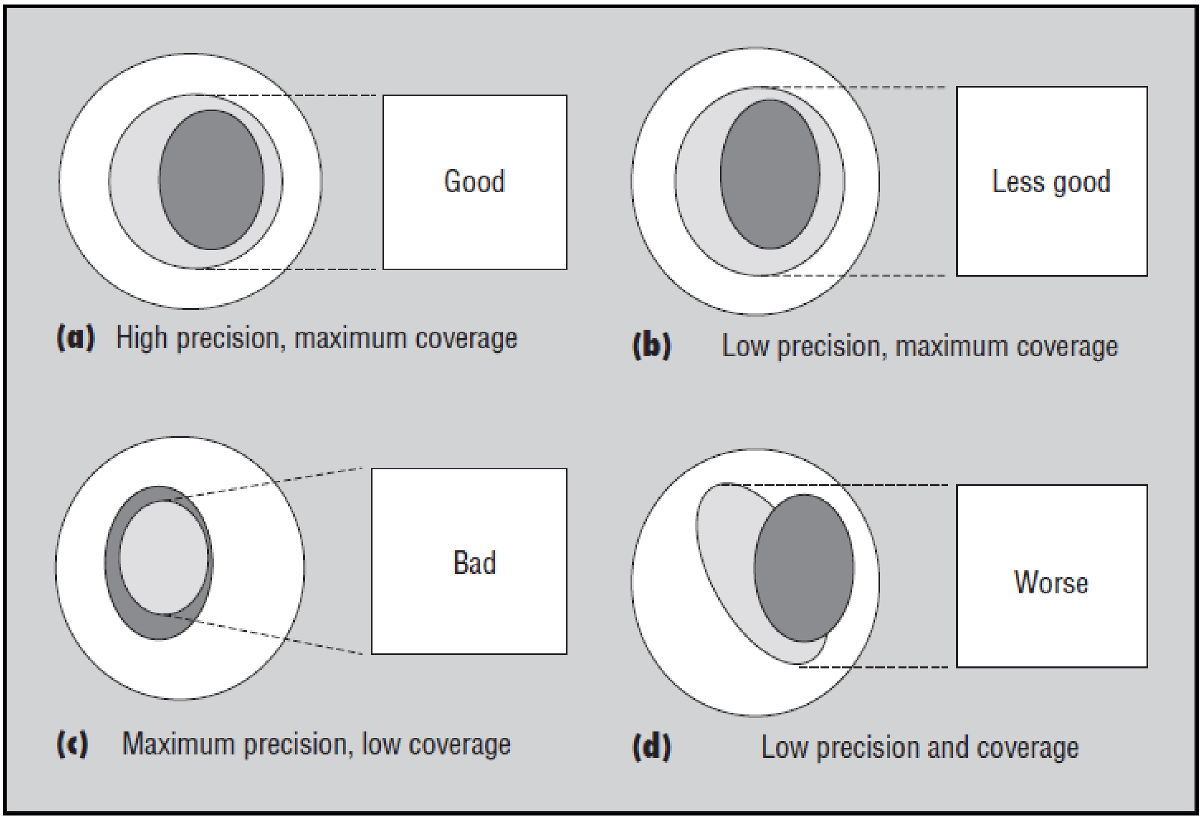

Figure 2-3: Ontology quality: different degrees of coverage and precision

Situation 1 isn't very interesting; we would say in this case that the ontology is totally "wrong" with respect to the particular conceptualization.

Situation 2 is an ideal case, which is almost impossible to reach.

Figure 2-3 shows Situations 3 through 5 and introduces the first two characteristics usable to formally evaluate an ontology: *coverage* and *precision*. Assuming that the domain *D* is finite (which implies that all the model sets in the figure are finite), we can define them as:

- $C = \dfrac{|I_K \cap O_K|}{|I_K|}$ (coverage)
- $P = \dfrac{|I_K \cap O_K|}{|O_K|}$ (precision)

Clearly, coverage is important for an ontology; if it goes under 100 percent, some intended model isn't captured.

Precision is often less important, especially if a certain user community knows in advance the meaning of the terms described by the ontology (as said above). However, imprecise ontologies can generate serious problems in cases where it's necessary to check whether two concepts are disjoint (ontology integration problem), as it will be explained here below.

Consider Figure 2-4 [Staab, 2004], which you can read in two ways.

In the first reading, assume that, $\mathbf{I}_{\mathbf{K}}(A)$ and $\mathbf{I}_{\mathbf{K}}(B)$ denote the set of all the possible instances of the two concepts *A* and *B*, that is, their possible intended interpretations under the commitment **K**. In this example, the two concepts are disjoint by hypothesis. However, if the ontology **O** is (more or less) imprecise, it might allow an overlap in the extension of the two concepts, as in the example shown in figure. So, practically speaking, the ontology **O** "believes" that *A* and *B* can have common instances.

The second possible reading denotes a worse situation, i. e. when imprecise ontologies that have different commitments, say $\mathbf{K}_A$ and $\mathbf{K}_B$, align themselves. Assume that the outside circle denotes the set of all possible models of a certain language **L**, while **O**(*A*) and **O**(*B*) are model sets of two different (rather imprecise) ontologies, relative to the same language **L**. Because of their imprecision, the two ontologies could have some models in common, indicating that they agree on something, but this might be <u>a false agreement</u> because no intended models are involved. So, we might risk relying on the two ontologies' syntactic interoperability, with no warranties concerning the actual intended meaning of the terms they define.

This is the reason why the so-called lightweight ontologies can't generally guarantee interoperability, and why axiomatic theories have to be developed based on "deep" ontological principles.

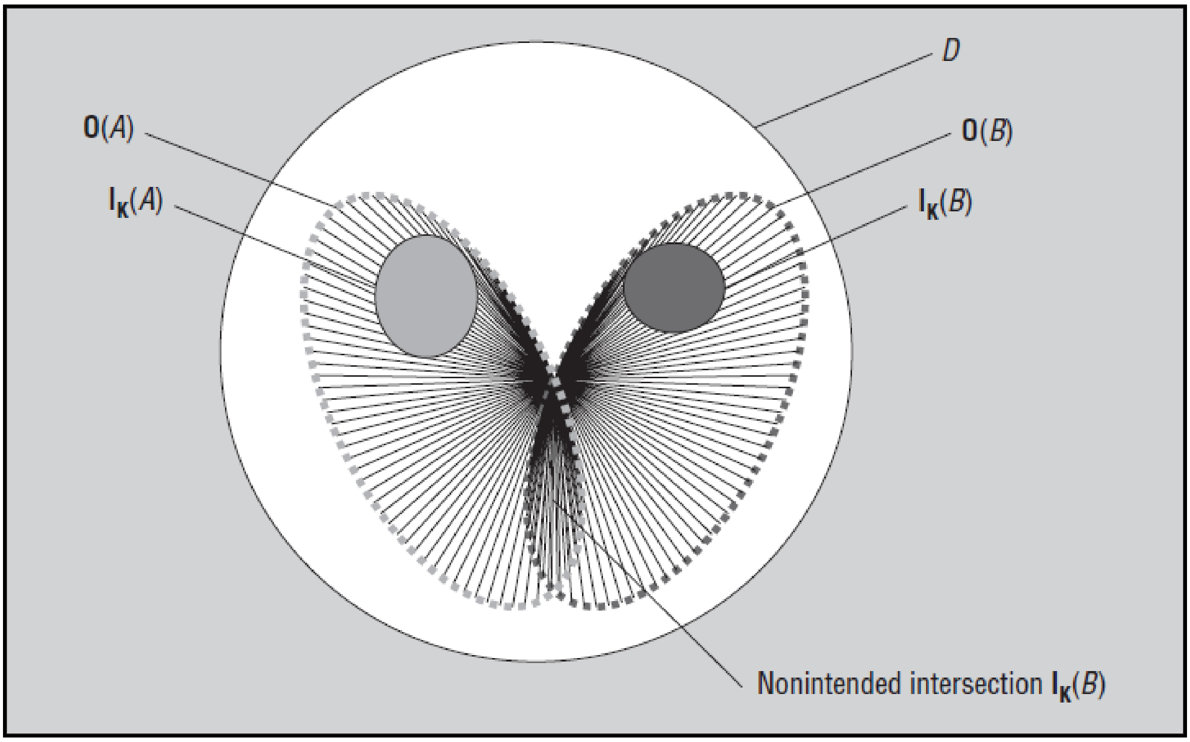

Figure 2-4: Imprecise ontologies and the risk of "false agreements"

The considerations above suggest that a bottom-up approach to systems integration based on the integration of multiple local ontologies may not work, especially if the local ontologies are only focused on the conceptual relations relevant to a specific *context*, and therefore they are only weak and *ad hoc* approximations of the intended models.

Hence, it seems more convenient to agree on a single *top-level* ontology rather than relying on agreements based on the intersection of different ontologies, and to develop different kinds of ontology according to their level of generality, as shown in Figure 2-5 [Guarino, 1998]:

- *Top-level ontologies*, which describe very general concepts like space, time, etc., which are independent of a particular problem or domain. It seems therefore reasonable, at least in theory, to have unified top-level ontologies for large communities of users.
- *Domain ontologies* and *task ontologies*, which describe, respectively, the vocabulary related to a generic domain like medicine or automobiles (it represents the particular meanings of terms as they apply to that domain), or to a generic task or activity like selling, by specializing the terms introduced in the top-level ontology.
- *Application ontologies* describe concepts depending both on a particular domain and task, which are often specializations of *both* the related ontologies.

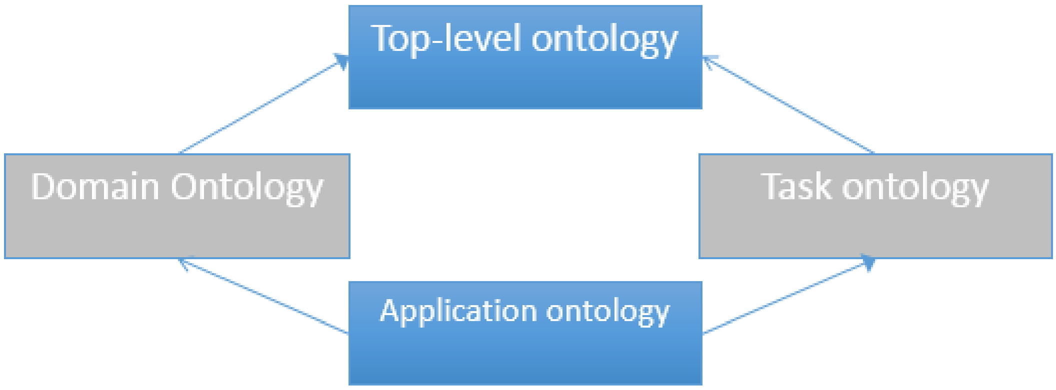

Figure 2-5: Categorization of Ontologies

2.4 Building An Ontology

2.4.1 Principles For Design

In order to represent something in an ontology, it's needed to make design decisions. To guide and evaluate these choices, objective criteria that are founded on the purpose of the resulting artifact, rather than based on a priori notions of naturalness or Truth, are needed.

Here below a preliminary set of design criteria for ontologies is presented, whose purpose is knowledge sharing and interoperation among programs based on a shared conceptualization [Gruber, 1993 b]:

1. Clarity: an ontology should effectively communicate the intended meaning of defined terms, i. e., definitions should be *objective*: while the motivation for defining a concept might arise from social situations or computational requirements, the definition should be independent of social or computational context.

   Formalism is a means to this end, hence:

   a. when a definition can be stated in logical axioms, it should be;

   b. where possible, a *complete* definition (a predicate defined by necessary and sufficient conditions) is preferred over a partial definition;

   c. all the definitions should be documented with natural language.

2. Coherence: an ontology should be coherent, that is, it should sanction inferences that are consistent with the definitions (at the least, the defining axioms should be logically consistent). Coherence should also apply to the concepts that are defined informally, such as those described in natural language documentation and examples. If a sentence that can be inferred from the axioms contradicts a definition or an example given informally, then the ontology is incoherent.

3. Extendibility: an ontology should be designed to anticipate the uses of the shared vocabulary. It should offer a conceptual foundation for a range of anticipated tasks, and the representation should be crafted so that one can extend and specialize the ontology *monotonically*. In other words, one should be able to define new terms for special uses based on the existing vocabulary, in a way that does not require the revision of the existing definitions.
4. Minimal encoding bias: the conceptualization should be specified at the knowledge level without depending on a particular symbol-level encoding. An encoding bias results when a representation choices are made purely for the convenience of notation or implementation. Encoding bias should be minimized, because knowledge-sharing agents may be implemented in different representation systems and styles of representation.
5. Minimal ontological constraint: an ontology should make as few claims as possible about the world being modeled, allowing the parties committed to the ontology freedom to specialize and instantiate the ontology as needed.

As understandable, ontology design will require making tradeoffs among the criteria. However, the criteria are not inherently at odds. For example, in the interest of clarity, definitions should restrict the possible interpretations of terms; minimizing ontological constraint, however, means specifying a weak theory, admitting many possible models. And yet, these two goals are not in opposition: the clarity criterion talks about definitions of terms, whereas ontological constraint is about the conceptualization being described: having decided that a distinction is worth making, one should give the tightest possible definition of it.

2.4.2 Ontology Development Methodologies

Two main groups of different development methodologies can be identified:

- The experience-based methodologies, such as the methodology proposed in [Grüninger and Fox, 1995], based on TOVE Project, or the other exposed in [Uschold and Gruninger, 1996] from Enterprise Model. Both were issued in 1995 and belong to the enterprise modeler domain.
- The methodologies that propose evolutive prototypes models, such as "METHONTOLOGY" [Gómez-Pérez et al., 2004], that proposes a set of activities to develop ontologies based on life cycle and the prototype refinement, and "101 Method" [Noy and McGuinness, 2001] that proposes an iterative approach to ontology development.

Obviously, there is not just one correct way or methodology for developing ontologies. Usually, the first ones are applied when the requirements are clearly known at the beginning, while the second ones are often used when the objectives are not well defined at the initial stage of the work. Moreover, it is common to merge different methodologies since each of them provides design ideas that distinguish it from the others.

However, in general terms, the ontology development can be divided into two main phases: specification and conceptualization. The goal of the specification phase is to acquire informal knowledge about the domain, while the goal of the conceptualization phase is to organize and structure this knowledge using external representations that are independent of the implementation languages and environments.

A possible way to adapt several ontology development methodologies and to define the specification and conceptualization phases has been provided in [Brusa et al., 2006], where the following steps to build an ontology are proposed:

-Specification phase

1. Defining ontology goal and scope

   This is an important step for minimizing the amount of data and concepts to be analyzed. The scope limits the ontology, specifying what must be included and what must not.

2. Drawing up a domain description
3. Defining motivation scenarios and competency questions

   The motivation scenarios have to show problems that arise when people need information that the system does not provide. Besides, the scenario description has to contain a set of solutions to these problems that includes the semantic aspects to solve them.

   Competency questions proceed from motivation scenarios. This allows deciding the ontology scope to verify if it contains enough information to answer these questions and to specify the detail level required for the responses. Besides, it defines expressivity requirements for the ontology because it must be able to give answers using its own terms, axioms and definitions. Finally, due to the fact that the scope must define all the knowledge that should be in the ontology as well as those that should not, a concept must not be included if there is not a competency question that uses it. This rule is also used to determine whether an axiom must be included in the ontology or not.

4. Defining ontology granularity and type according to the level of conceptualization and granularity shown in [Gómez-Pérez et al., 2004]

-Conceptualization phase

5. Defining a domain conceptual model
6. Identifying the ontology main components:
   a. classes: represent concepts, which are taken in a broad sense
   b. attributes: describe the classes in the ontology
   c. relationships: make explicit the link between classes
7. Defining ontology instances

2.5 Technical Realization

2.5.1 Languages

Ontology languages are formal languages used to encode ontologies.

There is a wide variety of such languages, both proprietary and standard-based, which can be classified as [Maniraj and Sivakumar, 2010]:

1. Logical Languages
    - First-order predicate logic
    - Rule based logic
    - Description logic
2. Frame based Languages (similar to relational databases)
3. Graph based Languages

In the following some among the most important ontology languages are introduced:

- **CL** (Common logic): is a framework for a family of logical languages, based on first-order logic, intended to facilitate the exchange and transmission of knowledge in computer-based systems.
- **CycL**: is the first ontology language ever created in computer science and artificial intelligence, and the one used by Doug Lenat's Cyc artificial intelligence project.

  The original version of CycL was a frame language, but the modern version is not. Rather, it is a declarative language based on classical first-order logic.

  There is a close variant of CycL known as MELD.
- **Gellish**: is a formal language that is natural language independent, although its concepts have 'names' and definitions in various natural languages. Indeed, information and knowledge are expressed in such a way that it is computer-interpretable, as well as system and natural language independent: all expressions, concepts and individual things are represented in Gellish by (numeric) Unique Identifiers (Gellish UID's).

  Each natural language variant, such as Gellish Formal English, is a controlled natural language (a structured subset of the natural one) and is suitable for information modeling and knowledge representation in this particular way.
- **KIF** (Knowledge Interchange Format): is a computer-oriented language for the interchange of knowledge among disparate computer programs.

  It has declarative semantics (i. e.: the meaning of expressions in the representation can be understood without appeal to an interpreter for manipulating those expressions), and it is logically comprehensive (i. e.: it provides for the expression of arbitrary sentences in the first-order predicate calculus).

  Although the original KIF group intended to submit it to a formal standards body, that did not occur. A later version (the above mentioned CL) has since been developed for submission to ISO and has been approved and published.

- **OWL** (Web Ontology Language[1]): is a family of knowledge representation languages or ontology languages for authoring ontologies or knowledge bases.

  OWL is endorsed by the World Wide Web Consortium (W3C) and has attracted academic, medical and commercial interest.

2.5.2 Tools

Earlier research on ontology design methodologies shows that manual construction of ontology is a complex process and it is very hard for a designer to develop a consistent ontology [Raja Mohan and Arumugam, 2005].

On the contrary, ontology editors tools enable the users for inspecting, browsing, codifying, and modifying ontology, and easily support, in this way, the ontology development and maintenance task. Existing editors vary in the complexity of the underlying knowledge model, usability, scalability, etc. (for instance, some tools support validation, others focus on translation, e. g.: UML into OWL), nevertheless, all of them provide enough support for the initial ontology development.

OntoEdit, OilEd, WebODE and Ontolingua are some examples of ontology editor [Raja Mohan and Arumugam, 2005], but the most common tool for ontologies is the Protégé Ontology Editor[2] with the Stanford University OWL-Plugin (there is also an Eclipse Plugin) [CRYSTAL_D_308_010].

[1] Please read http://www.w3.org/OWL/ for more information.
[2] Please read http://protege.stanford.edu/ for more information.

3 Rail Domain Ontologies

3.1 Ontologies As A Data Standard For The Rail Industry

3.1.1 Issues Affecting Data Sharing In The Rail Sector

The creation of an integrated European railway area calls for improved “interoperability” - or technical compatibility - of infrastructure, rolling stock, signalling and other subsystems of the rail system.

Promoting interoperability and harmonizing technical standards within the rail domain is the main challenge for EU Member States and rail stakeholders, and turns out in the greater challenge of integrating a large amount of data coming from several and heterogeneous sources.

The tasks of storing, processing and presenting the needed information are made more complex by a number of key factors [Easton et al., 2010]:

1) Legacy systems

   Both the infrastructure operators, and various train operating companies have some ability to monitor the condition of their assets. These systems provide useful information and they must form an important part of the smarter railway. Unfortunately, many of the current monitoring systems were developed and installed in isolation, and may also be operated by third-parties. Hence, the raw data, if available at all, are in proprietary formats, greatly increasing the complexity of any attempt to combine them with data from other sources.

2) Competing stakeholders

   Several stakeholders groups have different interests in the railways networks throughout Europe. Some of them, such as the train operating companies, may be in competition with each other and be unwilling to share information.

3) Differences in nomenclature and units of measurement

   While the use of terms within the railway industry in a particular country is usually consistent, this is not necessarily the case when considering railways in different parts of the world.

   While ambiguities of this type can usually be overcome by humans, it can be a very significant problem when automatically exchanging data between computer systems: for example, an XML file with a tag “shunt” generated in Britain (where this term indicates an operation when coaches or trucks are moved from one track to another, usually to change the formation of a train), may not have the same meaning as the tag in a file that originated in the US (where the equivalent term for “shunting” is “switching”).

   Finally another important point when considering the differences between railway networks of different countries lies in units of measure: Performing a conversion could be often necessary.

3.1.2 A Solution: The Adoption Of An Ontology-Based Data Standard

As understandable from the previous section, something must be done to improve data exchange and sharing within the rail industry, in order to bring about the so-called integrated European railway system.

As described in [Easton et al., 2010], an ontology-based data exchange standard has several advantages to offer:

-For railway industry

- Data transferred according to an ontology make it possible, for machines, to reason on them.

  By transferring condition monitoring data according to an ontology model, ontological inference can be used to generate a consistent set of data items from vehicles with differing instrumentation sets. From this, it should be much easier to generate metrics for the prediction of faults, since more data of a consistent form should be available to train the classifier being produced. This ability to predict failure of vehicles should lead to a reduction of in-service failures, and, obviously, thanks to this aid, the scheduled maintenance processes, which are costly and potentially unnecessary, could be safely reduced in frequency.

- Data transferred according to an ontology make it possible, for intelligent software agents, to perform a wide range of data processing tasks autonomously.

  Individuals with different roles within the industry may require radically different views of the information available to perform their assigned tasks. Data volume is also an important issue: the information must be filtered and summarized so that humans, who are ultimately responsible for the decision-making process, are not overloaded with information that is not relevant. In a system where data are transferred according to an ontology model, the combination of elements for particular tasks could ultimately be performed by software agents, making the software applications themselves simpler and more easily maintained.

- Ontologies could be easily extended by individual companies to meet their own needs, an important consideration when they are in competition with each other and may need to protect elements of intellectual property.

-For Interoperability

- The issues of nomenclature, units of measurement, etc. outlined above make an ontology-based data transfer standard essential for the development of an integrated European railway system.

- Railway scheduling presents a clear need for a common mechanism for sharing routes and scheduling information in a timely and unambiguous manner.

  Railway scheduling on a national network is a complex task: among the factors that must be considered are demand for a service, peak travel times, track and vehicle maintenance, speed limits, load limits, connecting services, timings of other services on the same line, and overall journey times. The situation is even more complex when considering cross border routes: in this case, the individual responsible for scheduling the service must have information on the railway network in each country the proposed service will pass through.

-<u>For Customer Experience</u>

- Software that communicates using an ontology can interact with other packages as part of the semantic web.

  Even though many of benefits outlined above, of course, impact on the customer experience (interoperability between networks, improved reliability, etc.), the most marked improvements will come through the rise of semantic web. In the traditional railway information system, in fact, the users have to create many queries to retrieve specific train information, and this problem can be effectively handled through semantic technology: semantic web, future web, will revolutionize the world with machine knowledge processing capabilities: indeed, semantic web is an extension of the current web in which information has a well-defined meaning -better enabling computers and people to work in cooperation-, and an ontology, which contains terms and relationships between terms, is its building block [Raja Mohan and Arumugam, 2005].

  Thanks to semantic web, for example, customers wishing to book train tickets to travel to a meeting, will only need to inform a software agent of their intended destination, and it will be able to find tickets for them; this might involve negotiating with the customer’s diary and interpreting the timetable to find the best time for their journey, arranging for overnight accommodation in their preferred hotel, and booking a table for dinner. Then, as the customer travels, agents will keep track of its progresses and will be able to adjust its itinerary if there will be a delay or a missed connection, updating the time of the meeting if that fits with the diaries of the other parties attending…

3.2 Creation Of A Railway Ontology

A number of sources of domain knowledge, already in existence within the rail industry, could be drawn on to speed the creation of an initial ontological model for the rail industry.

These include, but are not limited to, existing ontology models, normative standards, data exchange standards, and legislative guidelines.

Some of them (EN 50126, EN 50128 and EN 50129 CENELEC Standards, ISO 15926 standard, existing ontology models coming from InteGRail and CESAR projects, data exchange standards as RailML and MIMOSA OSA-CBM) will be introduced in the following sections (see Figure 3-1).

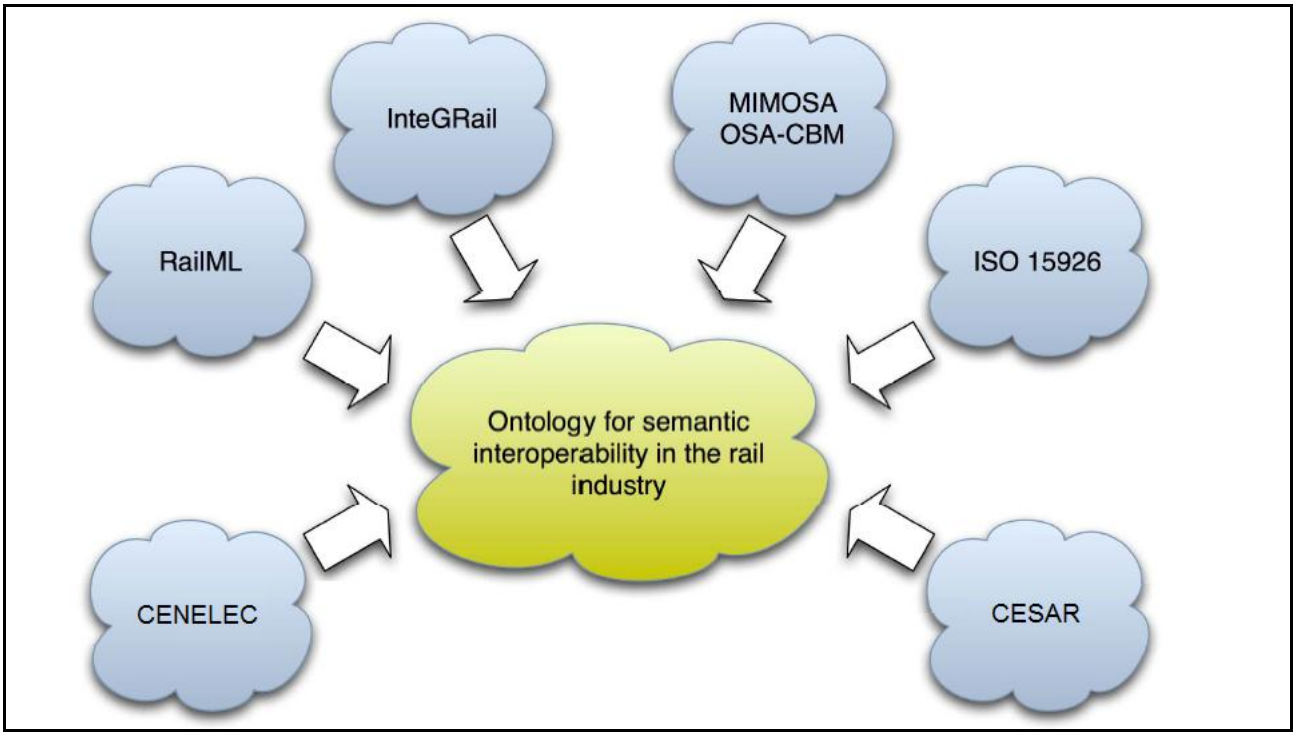

Figure 3-1: Sources of knowledge for a rail domain ontology.

3.3 Normative Standards

3.3.1 CENELEC Standards

The railway systems are critical systems since a failure can lead to disastrous consequences in terms of human lives or damages to the external environment, but also to economic penalties when availability and effectiveness levels reached by the system are not considered sufficient. As for other critical systems in other specific application domains, also in the railway field RAMS analysis and management appear in different standard. Those applicable in railway domain (in European context) have been produced by CENELEC, the European Committee for Electrotechnical Standardization, which is responsible for standardization in the electrotechnical engineering field. CENELEC produces standards and reference documents. CENELEC applies international standards, wherever possible, through its collaboration with International Electrotechnical Commission.

The EN 50126, EN 50128 and EN 50129 CENELEC standards represent the backbone of the RAMS demonstration process of a railway system: possible failures and hazards are identified during the overall lifecycle, they are properly corrected or mitigated considering their occurrence rate and the effort to spend; finally the risk is evaluated. In detail EN 50126 describes the processes and methods that are used to specify the most essential and important aspects for operability and safety in the rail domain; the EN50128 and the EN 50129 give a set of requirements which have to be satisfied during the safety-critical software (the former) / hardware (the latter) development, deployment and maintenance phases. The lifecycle suggested for these systems is a common ‘V’ lifecycle where the design is implemented during the descendent activities, which correspond to the verification and validation activities performed during the ascending branch.

As an example we report in Figure 3-2 the lifecycle suggested for software systems. The activities here reported are in relationships with the CENELEC EN 50128, where they are also detailed and discussed. On the contrary the CENELEC EN 50126 reports the lifecycle of the overall railway system with a particular focus on RAMS related activities.

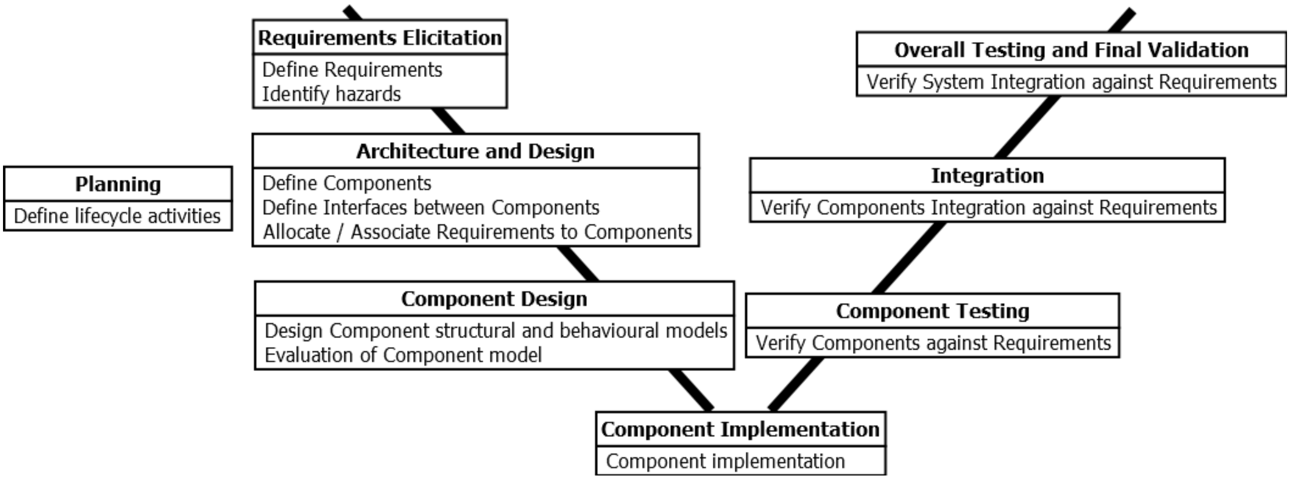

Figure 3-2: Lifecycle suggested for software artefacts (EN 50128)

The three standards also define the list and the minimal contents of the documents and deliverables that have to be produced during the lifecycle of a new railway system at system level, for software artefacts and for hardware subsystems. In particular they define how to manage all the steps spanning from concept to operation and maintenance, passing through design and installation.

Regarding ontological aspects, the three standards provide a set of definitions and abbreviations which shall be taken into account during the development of the Rail ontology. Example of these terms and abbreviations regards the following aspects:

- **RAMS features**: there are precise definitions of the RAMS characteristics (e. g.: availability, reliability, safety) and also of other aspects related to them (e. g.: accident, risk, failure, hazard);
- **Lifecycle**: they are clearly defined what some terms related to the main activity mean (e. g.: design, implementation);

- **Process**: it is clearly identified what some processes of a project shall perform (e. g.: hazard log, hazard analysis);
- **Roles**: the roles involved during all the phase are listed and precisely defined (e. g.: assessor, designer, implementer, integrator);
- **Others**: other minor terms are individuated and defined (e. g.: class of tools, differences between correction and repairs, etc.).

3.3.2 ISO 15926 Standard

The ISO 15926 standard was initially developed for the integration and exchange of information relating to process plants including oil and gas production facilities.

ISO 15926 takes a very “ground-up” approach to the modelling of processes, allowing each piece of equipment to be described in terms of its component parts as well as its temporal existence.

Recently, an OWL implementation of the ISO 15926 standard has been produced, which could serve as an upper-level ontology for the project [Easton et al., 2010].

3.4 Existing Ontology Models

3.4.1 InteGRail Project

InteGRail is a research Project addressing a wide number of coordinated objectives in the rail domain. InteGRail deals with railway information systems and their integration for improved overall efficiency and performance of the future European railway system. Inside this project a Railway Domain Ontology had been defined with a twofold objective: from one hand it provided a standard for information interchange between producers and consumers; from the other hand it represented an extensible mechanism for recognising context, creating opportunities for improved performance. The conceptual model was created using OWL. The proposed solution implements a semantically enabled network of reasoning nodes, where information is integrated and shared using the Railway Domain Ontology and distributed reasoning over service oriented architecture (SOA). The application of the Railway Domain Ontology aims at solving the integration challenge within the railway environment, achieving two main goals:

1. To have a standard and not ambiguous definition of term which can be interpreted also from computers.
2. To allow the automatic and powerful elaboration of information in order to process data and extract only those significant from them.

The results reached by the InteGRail project are reported in [UNIFE, 2010]; in the following we summarize the main goals reached for what concerning the Railway Domain Ontology.

The developed ontology would provide a generic solution for information interchange but it needs to be refined and validated. It should also be extended since according to railway needs, in order to cover new concepts, while maintaining and checking the consistency of the overall model. Furthermore the focus is on the rolling stock subsystem (that is only one of the main subsystems of a complete railway system) and specifically on vehicles. An extension is hence required in order to integrate signalling and on-board control system, which is the most complex and critical system of a railway.

3.4.2 CESAR Project

In the course of the CESAR project (Cost-Efficient Methods and Processes for SAfety Relevant Embedded Systems project), a first draft of a railway ontology, based on UNISIG (Union of Signalling Industry) SUBSET-026, which defines the current implementation of ETCS (European Train Control System) signalling equipment, has been provided by ASTS.

It is necessary to talk about a draft version because it covers only the needed requirements for the project itself and the related basic concepts (e. g.: the involved sub-systems), and hence it represents a very partial domain ontology.

However, after being shared among the main actors in railways, it could be considered as a starting point for the development on a complete domain ontology.

3.5 Data Exchange Standards

3.5.1 RailML

RailML is an XML schema which enables heterogeneous railway applications to communicate with each other. RailML is supported by an initiative founded in 2002 with the aim to simplify data exchange between railway applications. The RailML standard is not the result of a centralized work but it is steadily under discussion and in continuous development. In fact the last release of the RailML standard has been developed in June of 2013, while at the state last posts in the discussion forum are dated February 2014.

Between partners of the RailML initiative, three categories have been individuated: “RailML users” identifies the set of users who doesn’t active enhance the standard but these which daily exchange data in this format; “RailML developers” identifies the set of users who constantly creates RailML interfaces and suggest improvements to the standard; “RailML supporters” identifies the partners which don’t assure an active participation in the development of RailML schemes but advance the goal of this unified interfaces. At the state dozens of applications support the exchanging of data in accordance with the RailML standard.

The actual version of RailML, the 2.2, is organized in three subschemas and one other has been proposed:

- **Infrastructure (IS)**: this subschema contains the elements necessary to describe a complete railway infrastructure. In particular it encompasses data structures for describing track elements, switches but also tunnels and bridges. Also signals have been included such as data for displaying the specific topology;
- **Timetable and Rostering (TT)**: this subschema contains all data about any kind of timetables for both concept and operational lifecycle phase. This subschema is capable of providing primitives for describing also operating periods;
- **Rollingstock (RS)**: this subschema provides primitives for describing any kind of vehicles including locomotives, multiple units, passengers and freight wagons. It is also possible to describe an entire train as a fixed composition of a heterogeneous formation of vehicles.
- **Interlocking (IXL) – proposal**: at the state this subschema is under development and should contain information about typical interlocking data such as shunting routes, dependencies between signals, blocks and routes, switch and overlappings.

The RailML format can be easily imported in Eclipse according to the Ecore format. It is hence possible to adopt the EMF technology in order to develop a modelling editor which implements this standard as language.

3.5.2 MIMOSA OSA-CBM

MIMOSA (Machinery Information Management Open Standards Alliance) is an industry association, focused on enabling industry solutions leveraging supplier neutral and open standards, in order to establish an interoperable industrial ecosystem for Commercial Off The Shelf (COTS) solutions components provided by major industry suppliers.

Even though MIMOSA standards use their own data model, and may therefore not be suitable for a direct use within this context, among the family of standards they produce, there is OSA-CBM (Open System Architecture for Condition Based Maintenance), a data transfer architecture, designed (in UML) as a “multi-technological implementation”, being considered for use within the British railway industry [Easton et al., 2010].

4 Conclusions

An ontology-based standard for data transfer would have much to offer the rail industry, allowing existing, largely incompatible, legacy systems to exchange information in a meaningful way, without the need for costly changes that would potentially pose a risk to safety. More importantly, it would provide a framework on which modern, semi-autonomous processing agents could be built, improving the efficiency of the railway network, reducing the risk of human errors in mundane tasks, and enhancing the experience of the travelling public.

As said in the introduction, this section should provide a first evaluation of the above mentioned ontology catalogues in order to understand, above all, the gaps (interpreted as discrepancies) among the examined ontologies. Unfortunately, as a first evaluation of our analysis of the state of art in this field, we have to admit that the main gap (interpreted as lack of knowledge, this time) lies in the lack of influential and commonly-recognized railway ontologies which could be compared. Indeed, at the state, there are no noteworthy domain ontologies.

This is due to the fact that each railway operator works with its own ontology, and, even if all these ontologies are commonly founded on the adoptable standards and norms, there has not been in the past the possibility to accomplish a common result. Some attempts have been performed during some projects but, at the best of our knowledge, with poor results. Furthermore note that actual norms are applicable in specific countries (e.g., CENELEC norms are applicable only in European railway systems), and hence there is a lack of a worldwide applicable standard.

By thinking about the need of a domain ontology, for CRYSTAL in general (to optimize communication efforts among partners and improve the quality of the work packages results), and as a pre-requisite for defining IOS in particular, it's not hard to understand how it would be important that all the CRYSTAL partners within the rail domain collaborate together to work towards the creation of a unique ontology, which has to become the main communal objective during the next working months.

5 Terms, Abbreviations And Definitions

ASTS	Ansaldo STS
CENELEC	European Committee for Electrotechnical Standardization
CESAR	**C**ost-**E**fficient Methods and Processes for **SA**fety **R**elevant Embedded Systems
CL	Common Logic
CO	Confidential, only for members of the consortium (including the JU).
COTS	Commercial Off The Shelf
CRYSTAL	**Cr**itical S**YST**em Engineering **A**cce**L**eration
EMF	Eclipse Modelling Framework
ETCS	European Train Control System
IEEE	Institute of Electrical and Electronic Engineers
InteGRail	**INTE**lligent Inte**G**ration of **RAIL**way Systems
IS	Infrastructure
ISO	International Organization for Standardization
IXL	Interlocking
KIF	Knowledge Interchange Format
MIMOSA	Machinery Information Management Open Standards Alliance
OSA-CBM	Open System Architecture for Condition Based Maintenance
OWL	Web Ontology Language
R	Report
RailML	Railway Markup Language
RAMS	Reliability, Availability, Maintainability and Safety
RS	Rollingstock
RTP	Reference Test Platform
SOA	Service Oriented Architecture
SUO WG	Standard Upper Ontology Working Group
TOVE	**TO**ronto **V**irtual **E**nterprise
TT	Timetable and Rostering
UID's	Unique Identifiers
UML	Unified Modeling Language
UNISIG	Union of Signalling Industry
W3C	World Wide Web Consortium
XML	eXtensible Markup Language

Table 5-1: Terms, Abbreviations and Definitions

6 References

[Borst, 1997]	Borst W. N.; *Construction of Engineering Ontologies*; Centre for Telemática and Information Technology, University of Twente; 1997
[Business Week, 2002]	Business Week, 18/03/2002
[Brusa et al., 2006]	Brusa G., Caliusco M. L., Chiotti O.; *A Process for Building a Domain Ontology: an Experience in Developing a Government Budgetary Ontology*; in: Australasian Ontology Workshop (AOW 2006), Horbart, Australia; 2006
[Corcho, 2004]	Corcho O.; *A Declarative Approach To Ontology Translation With Knowledge Presetvation*; 2004
[Easton et al., 2010]	Easton J. M., Davies J. R., Roberts C.; *Railway Modelling - The Case for Ontologies in the Rail Industry*; KEOD 2010, pp. 257-262; 2010
[Genesereth and Nilsson, 1987]	Genesereth M. R., Nilsson N. J.; *Logical Foundation of Artificial Intelligence*; Morgan Kaufmann, Los Altos, California; 1987
[Gómez-Pérez et al., 2004]	Gómez-Pérez A., Fernández López M., Corcho O.; *Ontological Engineering with examples from the areas of knowledge management, e-commerce and the semantic web*; London: Springer; 2004
[Gruber, 1993 a]	Gruber T. R.; *A Translation Approach to Portable Ontology Specification*; Knowledge Acquisition 5 (2), pp. 199-220; 1993
[Gruber, 1993 b]	Gruber T. R.; *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*; in International Journal Human-Computer Studies 43, pp. 907-928; 1993
[Guarino, 1998]	Guarino N.; *Formal Ontology in Information Systems*; IOS Press, Amsterdam, pp. 3-15; 1998
[Guarino and Giaretta, 1995]	Guarino N., Giaretta P.; *Ontologies and Knowledge Bases: Towards a Terminological Clarification*; IOS Press, Amsterdam, pp. 25-32; 1995
[Grüninger and Fox, 1995]	Gruninger M., Fox M. S.; *Methodology for the Design and Evaluation of Ontologies*; IJCAI Workshop on Basic Ontological in Knowledge Sharing, Montreal, Canada; 1995
[Maniraj and Sivakumar, 2010]	Maniraj V., Sivakumar R.; *Ontology Languages – A Review*; International Journal of Computer Theory and Engineering, Vol. 2, No. 6, December, 2010, pp. 1793-8201; 2010
[Raja Mohan and Arumugam, 2005]	Raja Mohan A., Arumugam G.; *Constructing Railway Ontology using Web Ontology Language and Semantic Web Rule Language*; Int. J. Comp. Tech. Appl., Vol 2 (2), pp. 314-321; 2005
[Musen, 1992]	Musen M. A.; *Dimensions of knowledge sharing and reuse*; Computers and Biomedical Research 25, pp. 435-467; 1992
[Neches et al., 1991]	Neches R., Fikes R. E., Finin T., Gruber T. R., Senator T., Swartout W. R.; *Enabling technology for knowledge sharing*; AI Magazine 12 (3), pp. 36-56; 1991
[Nordland, 2003]	Nordland O.; *A critical look at the CENELEC Railway Application Standards*; presented at the TÜVIT seminar Application of the international standard IEC 61508, held in January

	2003 in Augsburg, Germany.
[Noy and McGuinness, 2001]	Noy N., McGuinness D.; *Ontology Development 101: A Guide to Creating Your First Ontology*; 2001
[ORG, 2007]	Ontology Research Group (ORG): http://org.buffalo.edu/OntologyDefs1.html; 10/2007
[Staab, 2004]	Staab S.; *Why Evaluate Ontology Technologies? Because It Works!*; IEEE Intelligent Systems 07-08/2014, pp.74-81
[Studer et al., 1998]	Studer R., Benjamins V. R., Fensel D.; *Knowledge Engineering: Principles and Methods*; IEEE Transactions on Data and Knowledge Engineering 25(1-2), pp. 161-197; 1998
[SUO WG, 2003]	Standard Upper Ontology Working Group (SUO WG): http://suo.ieee.org/; 28/12/2003
[UNIFE, 2010]	UNIFE - Association of European Railway Industries; *InteGrail – Publishable Final Activity Report*; 2010
[Uschold and Gruninger, 1996]	Uschold, M., Gruninger M.; *Ontologies: Principles, Methods and Applications*; Knowledge Engineering Review; 1996

Table 5-1: Terms, Abbreviations and Definitions References