The basics of interoperability: a curricula

Nacer Boudjlida, Hervé Panetto

The Basics of Interoperability: a Curricula

N. Boudjlida*, H. Panetto**

**Nancy Université, UHP Nancy 1, LORIA, UMR 7503*
Campus Scientifique, BP 70239
54506 Vandoeuvre Lès Nancy CEDEX, France
Nacer.Boudjlida@loria.fr

*** Nancy- Université, CRAN, UMR 7039, CNRS*
Campus Scientifique, BP 70239
54506 Vandoeuvre Lès Nancy CEDEX, France
Herve.Panetto@cran.uhp-nancy.fr

ABSTRACT: *In this paper, we introduce and we attempt to evaluate our experiment in lecturing the basics of interoperability. The lecturing includes an attempt to provide a formal foundation of interoperability, alternative approaches for model-based interoperability and related architectures, together with an introduction to semantic interoperability. A practical case study has been defined in order to put students into practice so they can discover some interoperability issues within an enterprise perspective.*

KEYWORDS: *Interoperability, Models and Meta-Models, Ontology, Model Transformation, Model Mapping, Semantic Interoperability.*

1. Introduction

In this paper we report on the experiment and the curricula to provide the basis for the interoperability domain to Master degree students, both with a research perspective and an application-oriented one. This course unit is offered at Nancy University (Nancy, France) and it is jointly part of a Master degree program in Computer Science, a Master degree program in Systems Engineering and a Master level program in an undergraduate engineering school in Computer Science. Some of the students are then pursuing their studies in a PhD program while others will get a job after the master degree. Therefore, the content of the lectures has been designed to encompass balanced parts: a theoretical part and an application-oriented one. This paper reports on this education course and it is structured as follows: sections 2 to 5 give an overview of the many facets of the curricula while section 6 concludes with a brief discussion on its application during four years.

2. Interoperability: Problem Statement and Formal Foundations

Among the variety of definitions of interoperability, we retain two main features that are the ability of two (or more) systems or agents (i) to exchange "things" (information, objects, rules, processes, etc.) and (ii) to cooperate to jointly carry out actions. The curricula firstly illustrates these features for simplicity, for fine grained information and data under different representations, considering a system S1 which produces a piece of information under a representation S1 (example: a character string information represented as the string value followed by an end-mark symbol) and which communicates that piece of information to a system S2 which admits that type of information under a representation S2 different from S1 (example: S2 admits the character string represented as the length of the string value followed by the string value itself). In this early introduction to the domain, S1 and S2 are assumed to "understand what they are exchanging". Therefore, we present the usual and well-admitted strategies for the achievement of the exchange: there are as many pairs of information/data converters (n*(n-1)) as the number *n* of interoperating systems or, alternatively, (2*n) converters when information/data transformation is performed thanks to an information format that is commonly admitted by the *n* interoperable systems. In order to give a theoretical basis for interoperability, the preceding situations are formally assimilated to *polymorphic functions* (Cardelli & al, 1985). This part of the lecturing typically requires some theoretical background together with some object-oriented programming background too.

The previous quite simple scenario enables the introduction of one of the main action that is usually performed for information/data exchange: data/information transformation.

3. Information Exchange for Interoperability

Generalizing information and data exchange for interoperability to less fine-grained information reveals the usual two-steps process: (i) transform the structure itself (for example, rewrite a date under the form *"dd-mm-yyyy"* into the form *"yyyy-mm-dd"* and (ii) possibly rewrite the value (for example, rewrite *mm* ∈ [1..12] into *mm* ∈ [Jan, Feb, …, Dec]). Then, we progressively go toward the discovery of the transformation process and the related problems that have to be solved (structural comparison, discrepancies detection and solving) (Sheth & al, 1993). Afterward, we attempt to give a formal framework for this process, considering a graph-based representation of information and data and considering two situations: (i) when dealing with strongly-typed data, we show that a kind of structural algebra can be defined for data transformation and (ii) when considering less strongly typed data (i.e. semi-structured data (Abiteboul & al., 2000)), the structural algebra is no more suited for this type of data: concepts and (abstract) algorithms are introduced for semi-structured data comparison and transformation.

Once these basics are introduced, concrete ways toward implementation are considered: alternative architectures are introduced and discussed. These include (i) systems encapsulation, (ii) mediator-based architectures, (iii) centralized or distributed, private or shared repositories, (iv) systems communication, (v) standard architectures and reference models of architectures taken from different domains (software engineering, object-oriented interoperability, workflow reference model, Model-Driven Engineering). These features typically refer to concepts and techniques coming from Computer Science (databases, distributed databases, client/server architectures and so on). Nevertheless, all what precedes emphasizes the need for the availability of accurate meta-data. However, till now, only structural meta-data was considered. So, when one no more assumes that two (or more) systems "understand" what they are exchanging, the need for additional type of meta-data becomes obvious and it leads to semantic interoperability.

4. Introducing Semantic Interoperability

Semantic interoperability requires introducing basic definitions of what is ontology and what are annotations, the later not being yet widely used in the interoperability domain. We demonstrate the role of semantic annotations in information exchange and transformation, distinguishing between (Boudjlida & al, 2007): (i) *structural annotations*, (ii) *lexical annotations* and (iii) *behavioural annotations*. Coming back to practical aspects, we conclude with the state of the art and the state of the practice concerning the many types of the automatic or semi-automatic support that can be brought (modelling support, ontology management support, and annotation support), and we introduce needs and approaches for interoperability solutions thanks to (meta-) meta-models.

5. Meta-modelling Languages for Interoperability

Within a model-based interoperability approach, the objective of the curricula is to improve students' knowledge and awareness about system engineering methods and tools, by integrating different models designed for complementary purposes (functional, behavioural, informational, …). We teach students on the various methodologies used to perform models integration, based on the formalisation of their respective definitions through meta-modelling. The UEML initiative, related to the development of a unique meta-modelling language (Panetto & al, 2004) is presented to demonstrate that a common language for business process modelling may be defined and used to exchange models between different enterprise modelling tools.

Finally, in order to put interoperability solutions into practice, the students exercise themselves against a case study where an exchange of technical data (Bills of Materials, Work In Progress orders) has to be performed between an ERP system and a MES. The data exchange must take advantage of an existing standard (IEC

62264) to develop a software mediator that will map information between both enterprise applications information models. However, the standard only relates to some model transformations without any semantic regarding the exchanged information. By analysing both applications data models and by using both applications on a real manufacturing process, the students can then study some semantics mappings between concepts in order to understand and to develop the main interoperability techniques developed in the curricula.

6. Conclusion and Evaluation

This course unit, related to the basics of interoperability, has been executed, every year, since 2004. The course audience was heterogeneous, not only in terms of students' background and skills but also in terms of future professional projects. The number of attendees was always around thirty. Around one third prepared a PhD degree after the Master degree and the others got a position in enterprises. Further, considering interoperability as a multi-disciplinary domain, the main problem that we had to address is having an understandable discourse for the variety of students' profiles. Indeed, the interoperability domain requires many skills, system engineering, system and data modelling, computer science, knowledge representation being among the ideally required skills. After four years, the current master degrees will be renewed and, starting in October 2009, Nancy-University will offer a research program (6 course units) at Master level on "System Engineering for Information Systems Interoperability", using video-conferencing and involving students and teachers from foreign universities.

7. References

S. Abiteboul, P. Buneman, and D. Suciu. *"Data on the Web; From Relations to Semi-Structured Data and XML"*. Morgan Kaufmann Publishers, San Francisco, CA, 2000.

N. Boudjlida and H. Panetto. *"Enterprise Semantic Modelling for Interoperability"*. Proceedings of the 12th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2007, 25-28 September, Patras, Greece.

L. Cardelli and P. Wegner. *"On Understanding Types, Data Abstraction and Polymorphism"*. ACM Computing Surveys, 17(4):471–522, December 1985.

Panetto H., Berio G., Benali K., Boudjlida N., and Petit M. *A Unified Enterprise Modelling Language for enhanced interoperability of Enterprise Models*. Proceedings of the 11th IFAC INCOM2004 Symposium, April 5th-7th, 2004, Bahia, Brazil, Elsevier

Sheth (A.P.) and Kashyap (V.). "*So Far (Schematically) Yet So Near (Semantically)"*. In Interoperable Database Systems. D.K. Hsiao, E.J. Neuhold and R. Sack-Davis Eds, North Holland, 1993.