Formal Fact-Oriented model transformations for Cooperative Information Systems semantic conceptualisation

Mario Lezoche, Alexis Aubry, Hervé Panetto

Formal Fact-Oriented model transformations for Cooperative Information Systems semantic conceptualisation

Mario Lezoche, Alexis Aubry, Hervé Panetto

Research Centre for Automatic Control (CRAN), Nancy-University, CNRS, Campus
Scientifique, Faculté des Sciences et Technologies, BP 70239, 54506 Vandoeuvre-lès-Nancy
Cedex, France
{mario.lezoche, alexis.aubry, herve.panetto}@cran.uhp-nancy.fr

Abstract: Information in enterprise is, now more than ever, a fundamental resource. In order to increase enterprise performance, economics paradigms focus on how to better manage it. Information Systems (IS) are systems whose activities are devoted to capture and to store data, to process them and produce knowledge, used by any stakeholders within an enterprise or among different networked enterprises. The modern architecture of information systems is based on distributed networks. An important challenge, to reach higher performance, is to represent and share knowledge managed by those ISs. One of the main issues in making such heterogeneous Cooperative Information Systems (CIS) working together is to remove semantics interoperability barriers. This paper firstly analyses interoperability issues between CISs and then proposes a systematic approach for data models conceptualisation for knowledge explicitation, based on initial conceptual model cleaning rules, expert knowledge injection rules and finally fact-oriented transformation rules. A case study is proposed, related to a work order process in an Enterprise Resource Planning application, Sage X3.

Keywords: Conceptual modelling, cooperative information systems, formal verification, data model conceptualisation

1   INTRODUCTION

The actual archetype for the Information Systems (ISs) involves large number of ISs distributed over large, complex computer/ communication networks. Such cooperative information systems (CIS) have access to large amount of information and have to interoperate to achieve their purpose. The cooperative information systems architects and developers have to face a hard problem: interoperability.

Interoperability can be defined as the ability for two or more systems to share, to understand and to consume information [20]. Some work [8] in the INTEROP NoE project has identified three different levels of barriers for interoperability: technical,

conceptual and organisational. Organisational barriers are still an important issue but out of scope of this paper. The technological barriers are strongly studied by researchers in computer science and the solution is generally based on models transformation [14].

Our research [24] focuses on the conceptual level of interoperability that is the ability to understand the exchanged information. A concept is a cognition unit of meaning [39], an abstract idea, a mental symbol. It is created through the action of conceptualisation, that is, a general and abstract mental representation of an object. During the history of human effort to model knowledge, different conceptualisation approaches regarding different application domains were developed [1].

This paper is dealing with a first step from a more general work focusing on the study of the semantic loss during the exchange of information representing business concepts. Quantifying the semantic gap between interoperating ISs implies enacting their semantics through their normalized conceptual models. Indeed, in this context, the starting point for semantics interoperability is related to models conceptualisation.

We will present a conceptualisation approach to make explicit the finest-grained semantics embedded into conceptual models for finally enabling two different information systems seamlessly interoperating.

Next section presents the general context of our work. Then, the following section details the fundamental pillars of our conceptualisation process. Then, we will propose a knowledge explicitation process starting from an implemented relational model to a fact-oriented conceptual one. This process allows us emphasizing the finest-grained semantics that must be enacted to study semantics interoperability between collaborating ISs.

Finally, to validate our proposal, a practical case study is presented based on an Enterprise Resource Planning application involved in a B2M (Business to Manufacturing) interoperation process.

2 COOPERATIVE INFORMATION SYSTEMS

Information Systems are systems whose activities are devoted to capture and to store data, to process them and produce knowledge, used by any stakeholders within an enterprise or among different networked enterprises. It is commonly agreed that Cooperative Information Systems provide a backbone for the Integrated Information Infrastructure [35]. Fully understanding and exploiting the advances in computing is the only way to encompass the complexity of constructing and maintaining such systems.

Although the progress made in information technology considerably improved the efficiency of applications development, its drawbacks and limitations are obvious and serious. In fact, the application models involved in a single application are numerous and different, each coping only with particular and partial aspects of the overall task. Moreover, the components technologies are heterogeneous, platform- and machine-dependant. The above-mentioned limitations and barriers measurably hinder the development and the maintenance process.

There is a growing demand to integrate such systems tightly with organizational work so that these information systems can be directly and immediately used by the business activity.

Here, the need of interoperation clearly appears. In fact, to achieve the purpose of the cooperation between the different Information Systems, information must be

physically exchanged (technical interoperability), must be understood (conceptual interoperability) and must be used for the purpose that they have been produced (conceptual and organisational interoperability). When trying to assess the understanding of an expression coming from a system to another system, there are several possible levels of interoperability [12]:

- *encoding:* being able to segment the representation in characters;
- *lexical:* being able to segment the representation in words (or symbols);
- *syntactic:* being able to structure the representation in structured sentences (or formulas or assertions);
- *semantic:* being able to construct the propositional meaning of the representation;
- *semiotic:* being able to construct the pragmatic meaning of the representation (or its meaning in context).

This tiered structure is arguable in general; it is not as strict as it seems. It makes sense because each level cannot be achieved if the previous levels have not been completed [12].

The encoding, lexical and syntactic levels are the most effective solutions for removing technical barriers for interoperability, but not sufficient, to achieve a practical interoperability between computerised systems. Dealing with trying to enable a seamless data and model exchange at the semantic level is still a big issue that needs conceptual representation of the intended exchanged information and the definition of the pragmatic meaning of that exchanged information in the context of the source and destination applications.

Different cooperation types have been investigated in ISO 14528 [21]. In fact, this standard considers that models could be related in three ways:

(1)   integration when there exists a standard or pivotal format to represent these models;
(2)   unification when there exists a common meta-level structure establishing semantic equivalence between these models; and
(3)   federation when each model exists per se, but mapping between concepts could be done at an ontology level to formalise the interoperability semantics.

Integration is generally considered to go beyond mere interoperability to involve some degree of functional dependence. Classifying interoperability problems [38] may help in understanding the degree of development needed to solve, at least partially, these problems but conceptualisation and semantics extraction is still an important issue because of the various contextual understanding of tacit knowledge embedded into those applications. The main prerequisite for achievement of interoperability of information systems is to maximise the amount of semantics which can be used and make it increasingly explicit [29], and consequently, to make the systems semantically interoperable. To highlight this issue, the paper is based on a referenced scenario involving enterprise systems applications.

Most of reverse engineering approaches [13] [9] return the information structure but present a model with tacit semantics. The ADM (Architecture-Driven Modernization) initiative [30] from OMG [5] is tackling this problem by promoting a common Knowledge Discovery Meta-model to facilitate discovering tacit knowledge embedded inside existing software. In our scenario, those applications are still implemented and running using databases. We can extract, from them, by using reverse engineering approaches, some knowledge in a form of a conceptual model. We have then to enrich that model with enterprise applications best practices

(knowledge coming from users). Finally, we make explicit all disclosed knowledge hidden in the resulting model.

3   OUR APPROACH FOR SEMANTICS ENACTMENT IN CONCEPTUAL MODELS

In order to cooperate, two (or more) Information Systems have to interoperate. As previously discussed, we focus our interest on the conceptual level of interoperability letting different information systems to share and use knowledge models that they represent. Our principal issues are, therefore, first to understand the conceptual relationships between those models in the context of their use and secondly how, through conceptualisation, to unhide the tacit knowledge buried inside them. A usual approach for making explicit the tacit knowledge, concealed in attributes and classes, is the relationships-oriented perspective composed of a set of transformation rules. In that transformation method, an attribute a_1 of type T_1 pertaining to a class C_1 is modelled as a relationship between the class C_1 and a standard type T_1. This approach does not resolve entirely the semantics elicitation problem because it focuses its point of view on the values instead of on the concepts. The attribute semantics is somewhat yet hidden in the relationship just created.

In literature, [27] presented the definition of two different objects types, *a lexical object (LOT),* a term*, is an object in a certain reality that can be written down. LOTs always consist of letters, numbers, symbols or other characters. They can be used as names for or references to other objects. A non-lexical object (NOLOT),* a concept, *is an object in a certain reality that cannot be written down. Non-lexical objects must be named by lexical objects or referred to by means of lexical objects.*

Applying these definitions, we can flatten the nested knowledge embedded in a model to simplify semantic enactment resulting from a set of modelling transformations. Our contribution is to have at our disposal an approach letting us to fragment knowledge through the transformation of attributes into entities and relationships, and thus to emphasize some fine-grained knowledge atoms. In the proposed approach, that is the first part (Figure 1) of our general methodology, the starting point can be various: an application, a data model, a logical view, a model. We have already mentioned that there are several reverse engineering methods, such as in [13] and in [9], through which a model from the application or schema level can be derived (Step 1). Then, the resulted initial model is enriched and corrected through an Expert Knowledge Injection step (Step 2). In fact, the model is examined with the help of a domain expert or an end-user, who are the most qualified persons to describe the context of the peculiar domain and to put in evidence the contextual knowledge. According to the enterprise best practices and its data, they would clean and better organise the knowledge represented in the derived model. However, the obtained initial conceptual model, in the form of a UML class diagram, has yet a major limit. In fact, its semantics is in a tacit form because all the attributes are buried inside single classes and it is then difficult to make their semantics explicit.
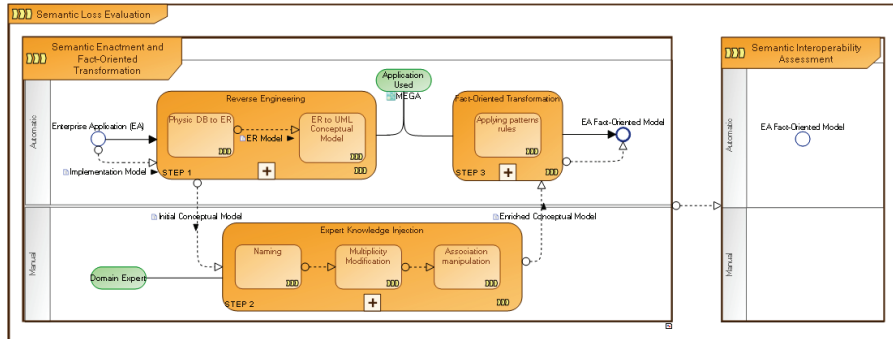
Figure 1 - Conceptualisation approach

Thus, the next step of our approach (Step 3) is a Fact-Oriented Transformation [16] through the application of a set of rules for transforming the enriched conceptual model to a fact-oriented model (FOM) with its semantics completely displayed. The consequence is that all the classes and their attributes are transformed into respectively LOTs and NOLOTs objects. The resulting fact-oriented model, displaying the finest-grained semantic atoms, is then used as an input for the second part of our methodology for semantic loss evaluation (not presented in this paper).

In the following sub-sections, we will discuss, in detail, the proposed 3 steps.

3.1   Step 1: Reverse Engineering

Conceptualisation is a decision process [15], a view, in which studied part of reality knowledge, usually in an implicit and complex form, is reorganised in different aggregates usually simpler to be represented.

According to [11], developing conceptual models means specifying the essential objects or components of the system to be studied, the relationships of the objects that are recognised and what kinds of changes in the objects or their relationships affect the functioning of the system and in which ways.

Conceptual models range in type from the more precise, such as the mental image of a familiar physical object, to the abstractness of mathematical models that do not appear to the mind as an image. Conceptual models also range in terms of the scope of the subject matter that they are taken to represent. The variety and scope of conceptual models is due to the variety of purposes that people had while using them.

Conceptualisation approaches are numerous and have been developed in different knowledge domains [23].

Our scenario assumes that we start from enterprise application database. So, the first studied approach is the Reverse engineering. It is, in database (DB) community, an approach to extract the domain semantics from the existing database structures. Typically, it concerns making the reverse transformation from logical to conceptual schema. In [13], the authors propose a general algorithm based on several old attempts to make explicit the logical structure buried into DB schemas, application programs and in the minds of designers and developers. [9] presents a methodology for extracting an extended Entity-Relationship model from a relational database, through a combination of data schema and data instance analysis. In our study we will consider at profit the reverse engineering experiences developed in the past. These

methods are, by now, acquired by the software industry that produces countless tools. We choose MEGA Suite (http://www.mega.com), a modelling management environment to transform relational models into conceptual ones. MEGA Suite implements a parameterised reverse engineering method coping with major existing approaches from direct database metadata analysis to a semi-automatic conceptual models building from existing database schemas.

3.2 Step 2: Expert Knowledge Injection

After the reverse engineering process has created a conceptual model, the current step concerns enriching it by injecting the enterprise knowledge, expressed by users' best practices or experts. These stakeholders know the domain peculiarities and they are capable to embed specific constraints into the new conceptual model. The first stage is the renaming process. Usually the database tables, and the derived concepts, have not standard names. The renaming process is essential to bring coherence and semantics in concepts that otherwise would be of very difficult comprehension. The following stage is the redefinition of the attributes and of the associations' roles multiplicities according to the enterprise users' best practices. This step is fundamental to define the real constraints that are not always made explicit into the implementation model. As an example, considering a particular attribute a_1, two cases can be considered:

1) a_1 is a non-mandatory attribute in the conceptual model but, as users are requested to always fill it with a specific value, the enriched model must formalise that this attribute a_1 is to be treated as mandatory;

2) a_1 is defined as mandatory in the conceptual model but, by practice, the users never care about its value and fill it with some dummy one. In such case, the enriched model may formalise that this attribute is not mandatory.

Note that the same cases may happen also to the roles of associations.

The last stage concerns of making explicit some implicit associations. Those implicit associations relate some concepts but they are defined only by enterprise practices even if they are not expressed in the model itself.

At this time, the enriched conceptual model formalises the whole application semantics (both the explicit one and the users' implicit one).

3.3 Step 3: Fact-Oriented Transformation

The quality of a conceptual model is often influenced by the conceptual language used for its specification. Most conceptual languages for data modelling are based on a version of Entity-Relationship modelling (E-R) [3] [10] [19]. However, these modelling languages are making a distinction between entities, attributes and relationships. On the contrary, in order to normalise the way that knowledge is represented, NIAM (Natural-language Information Analysis Method) [28] proposed to model the world in term of facts (either presenting terms (real things), or representing characteristics (attributes) of these real things), and relationships between facts. NIAM is attribute-free, it does not use explicitly the notion of attribute, treating all elementary facts as relationships. Some authors have extended the concepts and notations developed by NIAM with object orientation. It is the case of ORM (Object Role Modelling) [17]. Our purpose is to adapt this fact-oriented modelling approach to enriched conceptual models represented using the UML [31]

class notation. Thus, we developed a set of transformation modelling rules, to be applied to selected UML patterns (Table 1).

Let us refer to the definitions of LOT and NOLOT facts given in the beginning of section 3. Transforming a particular conceptual model in a fact-oriented model must follow these rules:

1. all classes are transformed into LOT facts. Using UML Class notation, a LOT fact is represented by a UML Class.
2. all attributes are transformed into NOLOT facts. Using the UML Class notation, a NOLOT fact is represented as a UML Class.
3. for each attribute *a* belonging to a UML Class *C*, an association is created between the corresponding LOT *a* and the corresponding NOLOT *C*, created by the two previous rules.
4. the multiplicity associated to each attribute *a* is copied as the multiplicity of the role of the previous (rule 3) association attached to the NOLOT *a*. The opposite role of the same association must have a constraint multiplicity equal to one.
5. all "simple" associations between classes are transformed into "simple" associations between NOLOTs.
6. all generalisation relationships between classes are transformed into "simple" associations with a constraint multiplicity equal to one on the role attached to generalised NOLOT and a non constraint multiplicity equal to * on the opposite role.
   In order to trace the fact that this association was coming from a generalisation, we annotate semantically the new corresponding association with a logical rule using OCL (Object Constraint Language) notation. Moreover, the inheritance feature of the generalisation association is mapped as new associations between LOTs representing the attributes of the generalised NOLOT, and all the specialised NOLOTs (sub-classes).
7. composition and aggregation relationships are transformed into simple association (rule 3) that keep unchanged the existing roles' multiplicities but trace their specific semantics through an attached semantic annotation formalised with an OCL logical rule.
8. association classes are transformed into a LOT fact with two associations linked to the corresponding initial LOT facts. The multiplicities of the roles of these two associations are determined inverting the ones initially formalised on the roles of the previous association.
9. any other specific constraints (generally modelled using OCL logical rules) are kept during the transformation process.
10. we did not take into account special cases of constraints in generalisations because they are not usually used in data conceptual modelling.

One of the conceptual modelling requirements is that a conceptual model must have formal foundations, which allow comparing that model with other conceptual models in a formal and exact way.

3.4   Patterns represented in FOL

[4] and [38] formalise UML class constructs semantics in First Order Language (FOL) axioms. We propose to adapt these works to formalise the fact-oriented model

patterns (presented previously) in FOL axioms.

We will present only one pattern rule formalisation in FOL: the "Class and Attributes" rule as presented in Table 1.

A class in UML designates a set of object with common features. Formally a *class C* corresponds to a FOL *unary predicate C.*

An *attribute a* of type *T* for a *class C* associates to each instance of *C* a set of instance of *T*, its multiplicity *[i..j]* specifies that *a* associates to each instance of *C* at least *i* and at most *j* instances of *T*. Formally, an attribute *a* of type *T* for *class C* corresponds to a binary predicate.

An association in UML is a relation between the instances of two or more classes. The multiplicity *[m..n]* attached to the role of a binary association specifies that each instance of the class *C* can participate at least *m* times and at most *n* times to the

Class and Attributes		Composite aggregation	
UML	FOM	UML	FOM
		Aggregation	
		UML	FOM
Generalisation		Association Class	
UML	FOM	UML	FOM

Table 1 - Fact-Oriented modelling patterns using UML notation

related association. An association *A* between two classes can be formalised as a binary predicate. In the studied pattern, we formalise a class C_1 containing two attributes A_1 and A_2 with respectively a multiplicity of 1 and [0..1], and with associated types respectively, A_1Type and A_2Type.

Its formalisation in FOL assertions is the following:

$$\forall x, y, \big(C_1(x) \wedge A_1(x,y)\big) \supset A_1 Type(y)$$

$$\forall x, z, \big(C_1(x) \wedge A_2(x,z)\big) \supset A_2 Type(z)$$

$$\forall x, C_1(x) \supset (1 \{y|A_1(x,z)\})$$

$$\forall x, C_1(x) \supset (0 \leq 1 \{y|A_1(x,y)\})$$

Applying the transformation rule, presented in section 3.3, to the class C_1, and to the two attributes A_1 and A_2, we will obtain the Fact-Oriented Model (FOM) in UML notation as shown in Table 1 "Class and Attributes".
Its formalisation in FOL assertions is the following:

$$\forall x_1, x_2, Assoc_1(x_1, x_2) \supset C_1(x_1) \wedge A_1(x_2)$$

$$\forall x_1, C_1(x_1) \supset (1 \{y|Assoc_1(x_1, y)\})$$

$$\forall x_2, A_1(x_2) \supset (1 \{y|Assoc_1(x_2, y)\})$$

$$\forall x_1, x_2, Assoc_2(x_1, x_2) \supset C_1(x_1) \wedge A_2(x_2)$$

$$\forall x_1, C_1(x_1) \supset (0 \leq 1 \{y|Assoc_2(x_1, y)\})$$

$$\forall x_2, A_2(x_2) \supset (1 \{y|Assoc_2(x_2, y)\})$$

We used as demonstration tools Prover9, an automated theorem prover for first-order and equational logic, and Mace4 searcher for finite models and counterexamples. We implemented our models in prover9 syntax to verify them in FOL formalised transformation models rules.

These artefacts are the constituent basis to represent the transformation rules in First Order Logic. After the translation, we own two sets of FOL expression facts. We named them A and B. They represent the formalised semantics of UML and FOM models. They can be used to verify the models semantic equality. We will use a standard verification algorithm based on set theory. The verification algorithm takes as input assumptions the two expression sets, A and B.

The goal of the verification task is to demonstrate $\forall x\ A(x) \wedge \neg B(x) \rightarrow \emptyset$.

With this evaluation method we are able to demonstrate that the semantics formalised in the initial conceptual model is equivalent or included into the one transformed in FOM and so it is sufficient for our verification purpose.

4 CASE STUDY

Interoperability between organisational and manufacturing activities is crucial in manufacturing enterprises. Production services have to produce, quickly and efficiently, the good product at the right moment. For this reason, they need at time information coming from others services, which need in return precise and update data on production.

We propose here to study and present the first part of such a B2M interoperability issue by considering a particular IS implemented in a real manufacturing environment: Sage X3 as an Enterprise Resource Planning (ERP) application.
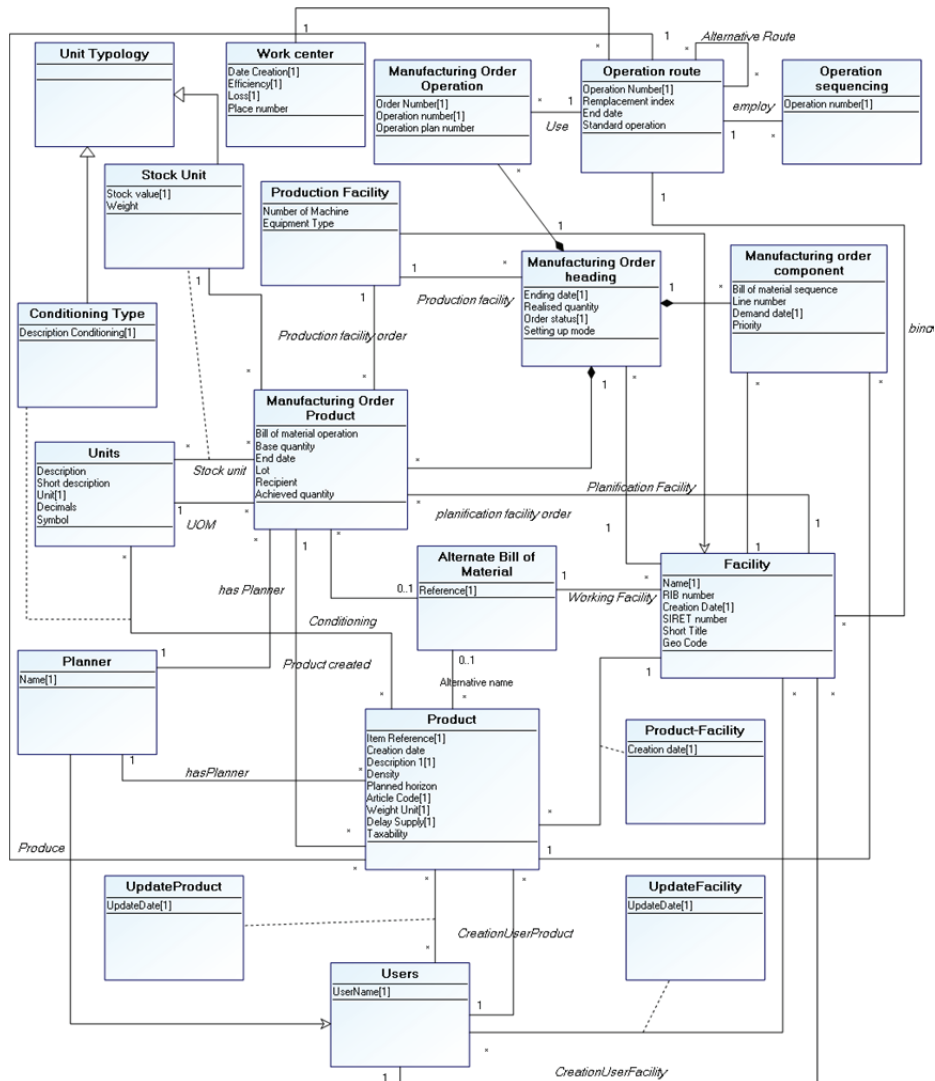

Figure 2 - Sage X3 work order enriched process model

4.1 Specific analysed Enterprise Information System: Sage X3 ERP

An Enterprise Resource Planning (ERP) is an integrated computer-based system used to manage internal and external resources including tangible assets, financial resources, materials, and human resources [6]. Its purpose is to facilitate the flow of information between all business functions inside the boundaries of the organization and manage the connections to outside stakeholders. Built on a centralized database, ERP systems centralise all business operations into a uniform system environment. Sage X3 presents different enterprise management functions: finance, commercial, industrial and services.

The focus for this case study is (i) to analyse how the work order process inside the Sage X3 application is modelled, (ii) to use the proposed modelling process to externalise the implicit knowledge in the model structure.

The model depicted in figure 2 is already the result from the two first steps of our approach. This means that we have already passed the "Reverse Engineering" and the "Expert knowledge injection" stages. The "Manufacturing Order Heading" concept is the management function of production orders and planned activities. It allows the generation of a production order by variation of one or more classifications and a single production line. For each production order, the achievement of the material benefits and sequencing operations is possible. This block captures general information about the work order, such as, planning facility and facility of production, status of the order (manufacturing order product). It allows entering general information about the production order. The availability of components is checked through the information given by the bill of material related with the launched products.

Once that initial information is determined, the system updates the list of materials and operations of the created or modified orders.

Step 1: Reverse Engineering

All these information are coded in the Sage application database. The first step of our method is the reverse engineering to extract the initial conceptual model.

Step 2: Expert Knowledge Injection

Currently the model depicted in Figure 2 is the result of the reverse engineering step enriched by a domain expert because the architecture of the Sage X3 ERP is built with all the database relationships implemented directly into the application layer and not in the database. The reverse engineering result, as shown in the lower part of the Figure 3, creates a model containing unlinked classes with coded names.

The expert work was about cleaning this conceptual model according to the best practices in the enterprise, modifying the attributes multiplicity, adding explicit names to the concepts, the attributes and the associations and others operations to fit the conceptual model to the "real" use of the Enterprise Information System. A usual case that requests the domain expert attention is about the mandatory properties in forms' attributes.

Step 3: Fact-Oriented Transformation

Applying the pattern transformation rules, presented in the previous section, class attributes are transformed into NOLOTs to increase the atomic representation of the knowledge embedded into the model. These rules have been coded using a programming language and then automatically executed inside MEGA Suite.

Figure 4 shows an extract of the resulting FOM after applying our approach to the Sage X3 work order process. The resulting full FOM is composed of 23 NOLOTs, 56 LOTs and 46 associations. It seems then that the resulting model is much more complex than the initial one, which it is true in a visual point of view but it is false in term of expressiveness of its semantics. Indeed, the fine-grained atoms of semantics are now made explicit, which helps any automatic computing. An important result is that such semantically detailed model will help automating the next part of our methodology for semantic gap evaluation, as presented in [41].

5   CONCLUSIONS

In this article, a conceptualisation approach for enacting implicit semantics from Enterprise Information Systems is proposed. Our approach if divided into 3 steps from the traditional reverse engineering process, through a knowledge elicitation and model enrichment by domain experts, till making use of fact-oriented modelling patterns to externalise tacit knowledge. These patterns have been formalised in FOL axioms to verify their semantic coherence. Our contribution can be assimilated to a reverse engineering methodology. However, the main objective is to formalize the whole semantics of such models in order to help automatic knowledge computing. An industrial case study, related to an enterprise information system implemented into an ERP system demonstrates the applicability of our approach.

Our current work concerns applying this approach for evaluating the (non)-interoperation through the measurement of the semantic gap occurring between CISs interoperability [40] and [41].
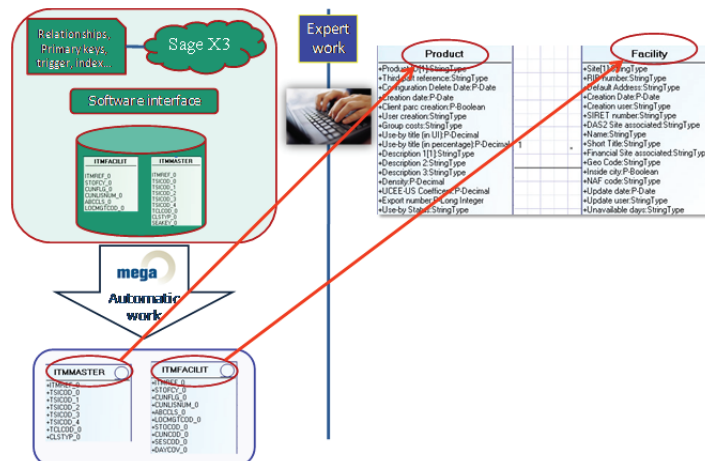

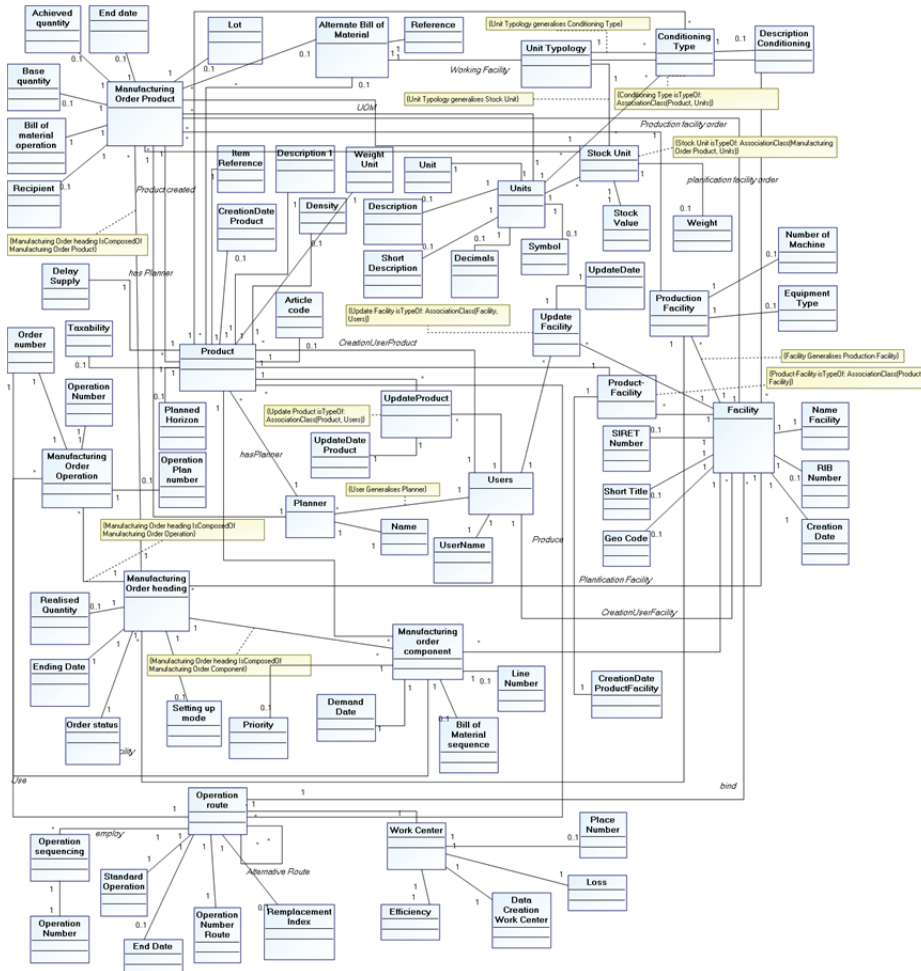Figure 3 – Sage X3 architecture and expert knowledge injection

Figure 4 – Sage X3 work order process model part transformed with fact-oriented approach

REFERENCES

[1]   Aspray, W. F. 1985. The Scientific Conceptualization of Information: A Survey. Annals of the History of Computing, 7: 117-40. IEEE

[2]   Badia, A., 2002. Conceptual Modeling for Semistructured Data. In *Proceedings of the 3rd International Conference on Web Information Systems Engineering Workshops* (WISE 2002 Workshops), p. 170-177. Singapore.

[3]   Barker, R. 1990, CASE* Method: Entity Relationship Modelling, Addison-Wesley, Wokingham, England.

[4]   Berardi, D., Calvanese, D., De Giacomo, G. (2005). Reasoning on UML class diagrams, Artificial Intelligence 168 (1–2) 70–118.

[5]   Bézivin, J., Kurtev, I. 2005. Model-based Technology Integration with the Technical Space Concept. Proceedings of the Metainformatics Symposium, Esbjerg, Denmark, November 8-11, 2005. Springer-Verlag

[6]   Bidgol, H., 2004. *The Internet Encyclopedia*, Volume 1, John Wiley & Sons, Inc. p. 707.

[7]   Carney, D., Fisher, D., Morris, E., Place P., 2005. Some current Approaches to Interoperability. In *technical note* CMU/SEI-2005-TN-033.

[8]   Chen, D., Dassisti, M., Elvesaeter, B., Panetto, H., et al., 2006. In *DI.2: Enterprise Interoperability Framework and knowledge corpus*, Interoperability Research for Networked Enterprises Applications and Software Network of Excellence, n° IST 508-011.

[9]   Chiang Roger, H. L., Barron, T. M., Storey Veda, C., 1994. Reverse engineering of relational databases: Extraction of an EER model from a relational database. In *Data & Knowledge Engineering*. Vol. 12, Issue 2, 107-142.

[10]   Czejdo, B., Elmasri, R., Rusinkiewicz, M. & Embley, D.W. 1990, 'A graphical data manipulation language for an extended entity-relationship model', IEEE Computer, March 1990, pp. 26-37.

[11]   Engelbart, D.C., 1962. Augmenting human intellect: a conceptual framework. In *Menlo Park, CA*: Stanford Research Institute.

[12]   Euzenat, J., 2001. Towards a principled approach to semantic interoperability. In *CEUR Proceedings of the IJCAI-01* Workshop on Ontologies and Information Sharing, Seattle, USA, August 4-5, , ISSN 1613-0073, Vol. 47., 19-25.

[13]   Fonkam, M.M., Gray, W.A., 1992. An Approach to Eliciting the Semantics of Relational Databases. In *CAiSE 1992*, Manchester, UK, May 12-15, 1992. Lecture Notes in Computer Science 593 Springer, ISBN 3-540-55481-5. 463-480 Manchester, UK.

[14]   Frankel D. S. 2003. Model Driven Architecture: Applying MDA to Enterprise Computing. John Wiley & Sons.

[15]   Guarino, N., 1998. *Formal Ontology in Information Systems* (Ed.) IOS Press.

[16]   Halpin, T.A. 1991, 'A fact-oriented approach to schema transformation', Proc. MFDBS-91, Springer Lecture Notes in Computer Science, no. 495, Rostock.

[17]   Halpin, T., 1998. *Handbook on Architectures of Information Systems* Chapter 4, eds P. Bernus, K. Mertins & G. Schmidt, Springer-Verlag, Berlin.

[18]   Henrard, L., Hainaut, J.-L., 2001. Data Dependency Elicitation in Database Reverse Engineering Software Maintenance and Reengineering. In *Fifth European Conference on Software Maintenance and Reengineering*, pp. 11

[19]   Hohenstein, U. & Engels, G. 1991, 'Formal semantics of an entity-relationship-based query language', Entity-Relationship Approach: the core of conceptual modelling (Proc. 9th ER conf.), ed. H. Kangassalo, Elsevier Science Pub., Amsterdam

[20]   IEEE: Standard Computer Dictionary, 1990. A Compilation of IEEE Standard Computer Glossaries. In *NY. 610-1990*. ISBN: 1559370793.

[21]   International Organization for Standardization, 1999. ISO 14528: Industrial Automation Systems – Concepts and rules for Enterprise Models, *TC 184/SC5/WG1*, Geneva, Switzerland.

[22]   International Organization for Standardization, 2002. ISO 16100: Manufacturing Software Capability Profiling for interoperability. In *Part 1: Framework, TC 184/SC5/WG4*, Geneva, Switzerland.

[23]   LaOngsri, S. 2009, Semantic Extensions and a Novel Approach to Conceptual Modelling, Ph.D. Thesis, School of Computer Science, Engineering and Mathematics, The Flinders University of South Australia

[24]   Lezoche M., Panetto H., Aubry A., 2011, Conceptualisation approach for Cooperative Information Systems interoperability. 13th International Conference on Enterprise Information Systems ICEIS 101-110, Beijing, P. R. China, DOI: 10.5220/0003508401010110.

[25]   Mani, M.: EReX, 2004. A Conceptual Model for XML. In *Proceedings of the Second International XML Database Symposium* (XSym 2004), 128-142. Toronto, Canada.

[26]   Manola, F., Miller, E., 2004. RDF Primer. In *World Wide Web Consortium*, Recommendation REC-rdf-primer-20040210.

[27]   Meersman, R., Tari, Z., 2003. Creating a "DOGMAtic" Multilingual Ontology Infrastructure to Support a Semantic Portal On The Move to Meaningful Internet Systems 2003. In OTM 2003 Workshops OTM Confederated International Workshops, Lecture Notes in Computer Science, Vol. 2889. ISBN: 978-3-540-20494-7, Catania, Italy.

[28]   Nijssen, G.M. & Halpin, T.A. 1989, Conceptual Schema and Relational Database Design, Prentice Hall, Sydney.

[29]   Obrst, L., 2003 Ontologies for semantically interoperable systems. In *Proceedings of the 12th International Conference on Information and Knowledge Management*. New Orleans, USA

[30]   OMG, 2003. Object Management Group. Architecture-Driven Modernization specification http://adm.omg.org

[31]   OMG, 2004. Object Management Group. UML 2.0 Superstructure Specification http://uml.omg.org

[32]   Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen W., 1991. Object Oriented Modeling and Design. *Prentice Hall*, Book Distribution Center, New York, USA.

[33]   Russel, S., Norvig, P. *Artificial Intelligence, A Modern Approach"*, Prentice-Hall. Inc. 1995

[34]   Seeley, R. S., 1997. Manufacturing execution systems in *MED DEVICE DIAGN IND*. Vol. 19, no. 11, pp. 64-68.

[35]   Sheth, A., 1998. Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics. In M. Goodchild, M. Egenhofer, R. Fegeas, and C. Kottman, editors. In *Interoperating Geographic Information Systems*, pp. 5– 30. Kluwer.

[36]   Smith, M.K., Welty, Ch., McGuinness, D.L., 2004. OWL Web Ontology Language Guide. In *World Wide Web Consortium*, Recommendation REC-owl-guide-20040210.

[37]   Tolk, A., Diallo, S. Y., Turnitsa, C. D., 2007. Applying the Levels of Conceptual Interoperability Modelling Support of Integrability, Interoperability, and Composability for System-of-Systems Engineering. In *Journal of Systemics, Cybernetics and Informatics*, Volume 5 Number 5, pp. 65-74.

[38]   Tursi, A., Panetto, H., Morel, G., Dassisti, M., 2009 Ontological approach for Products-Centric Information System Interoperability in Networked Manufacturing Enterprises. In *IFAC Annual Reviews in Control*. 33/2, 238-245, Elsevier, ISSN: 1367-5788.

[39]   Vyvyan, E., 2006. Lexical Concepts, Cognitive Models and Meaning-Construction. In *Cognitive Linguistics* 17 (4): 491-534.

[40]   Yahia E., Lezoche M., Aubry A., Panetto H. 2011. Semantics enactment in Enterprise Information Systems. 18th IFAC World Congress. Milan, Italy. IFAC PapersOnline.

[41]   Yahia E., 2011. *Contribution à l'Evaluation de l'Interopérabilité Sémantique entre Systèmes d'Information d'Entreprises : Application aux Systèmes d'Information de Pilotage de la Production*. PhD Thesis, Université Henri Poincaré, Nancy I (in French).