HAL
open science

Conceptualising and structuring semantics in Cooperative Enterprise Information Systems Models

Mario Lezoche, Esma Yahia, Alexis Aubry, Hervé Panetto, Milan Zdravković

To cite this version:

Mario Lezoche, Esma Yahia, Alexis Aubry, Hervé Panetto, Milan Zdravković. Conceptualising and structuring semantics in Cooperative Enterprise Information Systems Models. Computers in Industry, 2012, 63 (8), pp.775-787. 10.1016/j.compind.2012.08.006 . hal-00722419

HAL Id: hal-00722419

https://hal.science/hal-00722419

Submitted on 1 Aug 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Conceptualising and structuring semantics in Cooperative Enterprise Information Systems Models

Mario Lezoche[1,2*], Esma Yahia[1,2], Alexis Aubry[1,2], Hervé Panetto[1,2], Milan Zdravković[3]

[1]*Université de Lorraine, CRAN, UMR 7039, Boulevard des Aiguillettes B.P.70239, 54506 Vandoeuvre-lès-Nancy, France*

[2]*CNRS, CRAN, UMR 7039, France*
E-mail: {mario.lezoche, esma.yahia, alexis.aubry, herve.panetto}@univ-lorraine.fr

[3]*Faculty of Mechanical Engineering in Nis, University of Nis, ul. Aleksandra Medvedeva, 14, 18000, Nis, Serbia*
E-mail: milan.zdravkovic@gmail.com

Enterprise performance is, now more than ever, one of the key points for reaching the market success. In order to increase it, economics paradigms focus on how to better manage knowledge acquiring, sharing and update. Knowledge management can be approached with the possibility offered by the sustainability goals trying to optimize different enterprise strategic domains. The modern architecture of information systems (ISs) is based on distributed networks with a grand challenge of representing and sharing knowledge managed by ISs and consequently, to remove semantics interoperability barriers. First, this paper analyses interoperability issues between Cooperative Enterprise Information Systems (CEIS). Based on this analysis, the authors propose a conceptualisation approach for semantics discovery and management in Enterprise Information Systems models, based on applying fact-oriented transformation rules. The input of the transformation process is a conceptualised UML class model, reverse-engineered from an implemented model, and transformed into a Fact-Oriented Model (FOM), which makes explicit the finest-grained semantics. Semantics aggregates are then computed for structuring the whole semantics embedded in enterprise applications. They define independent set of concepts with their own minimal mandatory semantics. Finally a case study is proposed to validate the practicability of our approach in a real scaled scenario involving an Enterprise Resource Planning (ERP) and a Manufacturing Execution System (MES).

Keywords: Conceptual modelling, enterprise information systems, semantic interoperability, data model conceptualisation, UML, Fact-Oriented Model, model transformation, transformation rules.

1 - Introduction

The actual archetype for the Enterprise Information Systems (EISs) involves large number of ISs distributed over large, complex networked architecture. Such cooperative enterprise information systems (CEIS) have access to a large amount of information and have to interoperate to achieve their purpose. CEIS architects and developers have to face a hard problem: interoperability.

* Corresponding Author: mario.lezoche@univ-lorraine.fr

While networked systems engineering is more and more focusing on sustainable perspectives, new scientific advances in systems interoperability might contribute to make systems interoperability sustainable. In fact, it can improve sustainability themes on data, semantic, and structural mappings between partner enterprises in the complex network for creating a stable interoperable enterprise operating environment with long term exploitation.

Interoperability can be defined as the ability of two or more systems to share, understand and consume information (IEEE, 1990). (EIF, 2010) has identified three different levels of interoperability: technical, semantic and organisational. Organisational barriers are still an important issue but they are out of scope of this paper. The technological barriers are strongly studied by researchers in computer science and they are, in general, addressed by the models transformation (Frankel, 2003).

Our research focuses on the semantic level of interoperability, namely the ability to understand the exchanged information. Information may be defined as data linked to knowledge about this data. It is represented by so-called concepts. A concept is a cognitive unit of meaning (Vyvyan, 2006), a mental symbol.

We will contribute to the science foundation of the interoperability domain with a conceptualisation approach for making explicit the finest-grained semantics - the smallest possible piece of information - grouped together into atomic concepts. These so called atoms of semantics are extracted from conceptual models in order to facilitate the semantic matching between two different information systems that have to interoperate. In fact these atoms represent the found smallest possible piece of information to overcome the subjectivity in the modelling activity performed by models architects (Lezoche et al, 2012).

Moreover a knowledge structuring process is necessary for making explicit the semantic dependencies between the discovered concepts. This structuration is based on identifying

semantics aggregates (namely semantic blocks) that represent independent piece of knowledge containing their own minimal semantics (Yahia et al, 2012a).

Linking the works presented in (Lezoche et al, 2012) and (Yahia et al, 2012a), this paper proposes a complete process of conceptualisation and semantics structuration as a requirement for assessing interoperation capabilities when two or more companies want to cooperate. This model-based assessment contributes to the sustainability of the interoperability process by tracing, archiving and highlighting core concepts and concept aggregates necessary for improving the collaboration.

The semantic interoperability measure process is not part of this work but has been published in (Yahia et al, 2012b).

Next section presents the general context of our work, namely cooperative enterprise information systems. Then, we will present the knowledge explicitation process that transforms an implemented relational model into a normalised fact-oriented conceptual one. This resulting model makes explicit all atoms of semantics, most of the time hidden into the implemented model. In the fourth section, the semantics structuring process is described. It formalises the definition of semantics aggregates that highlight the structure of the embedded semantics in the conceptual model obtained after the conceptualisation process. Each semantics aggregate (namely, each semantic block) is associated with a concept and defines the minimal mandatory semantics attached to this concept. For validating our proposal, a practical case study is presented in the fifth section. It is based on two applications: an Enterprise Resource Planning (ERP) and a Manufacturing Execution System (MES), involved in a Business to Manufacturing (B2M) interoperation process. Finally, after discussing on the contribution of this work and its limits, we will conclude with some perspectives for future works.

2 - Cooperative Information systems

Information Systems are systems whose activities are devoted to capture, store and process data and to 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 (Sheth, 1998). Although the progress made in Information Technology (IT) considerably improved the efficiency of software development, the models involved in a single application development are numerous and diverse. Moreover, the components' technologies are diverse, platform- and machine-dependant. The above-mentioned limitations and barriers hinder the development and the maintenance process, significantly.

The main difficulty in analysing the detailed semantics of cooperative information systems models concerns the big amount of concepts and relationships, most of the times "polluted" by implementation constraints. Indeed, deeply studying the semantics embedded into a conceptual model such as the one represented in Figure 1 is really complex and hard to be done manually. Our proposed approach aims at automatically computing a structured semantic model (Figure 2) that makes easier this analysis.

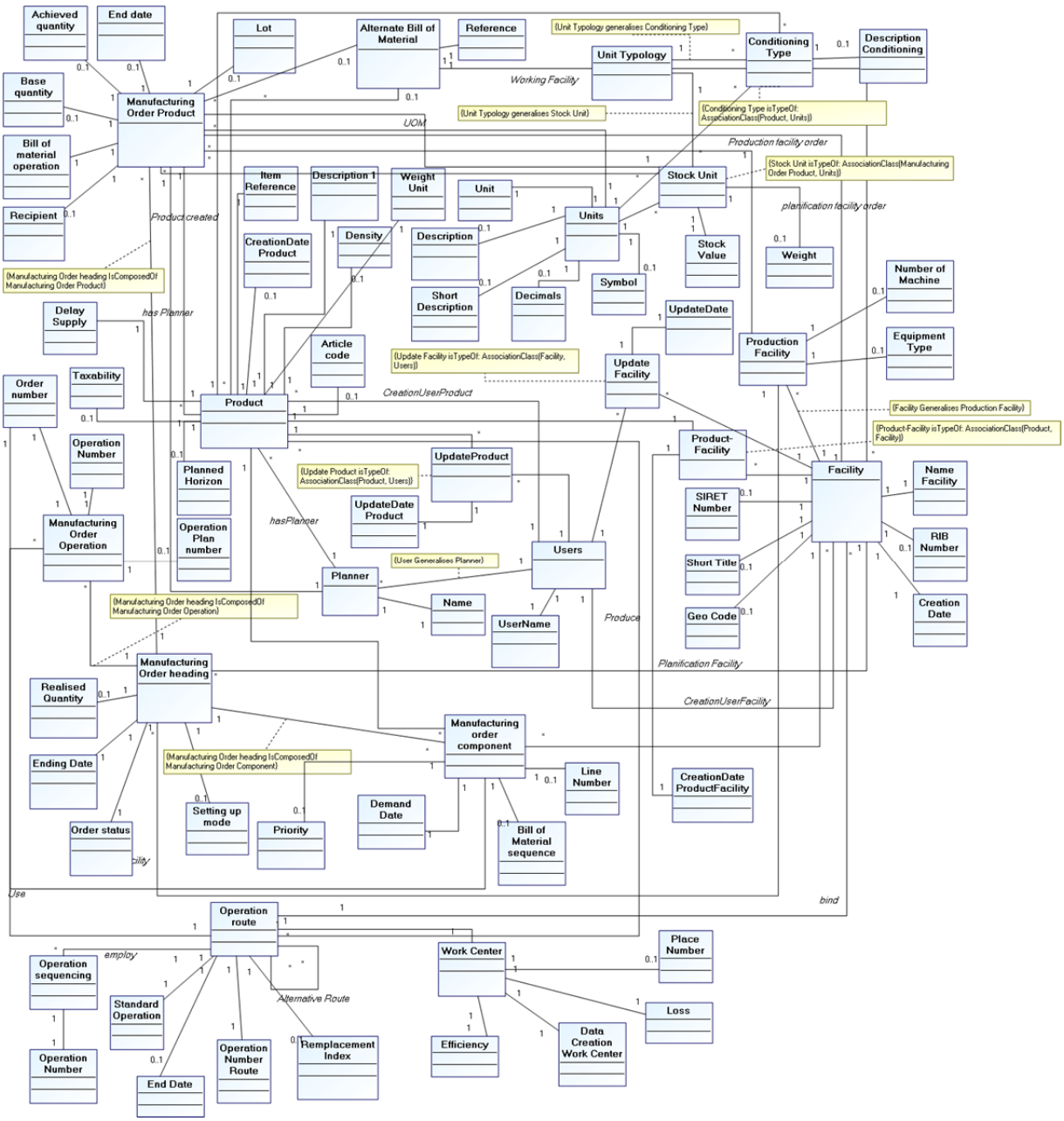


Figure 1 - ERP Sage X3 manufacturing order process model

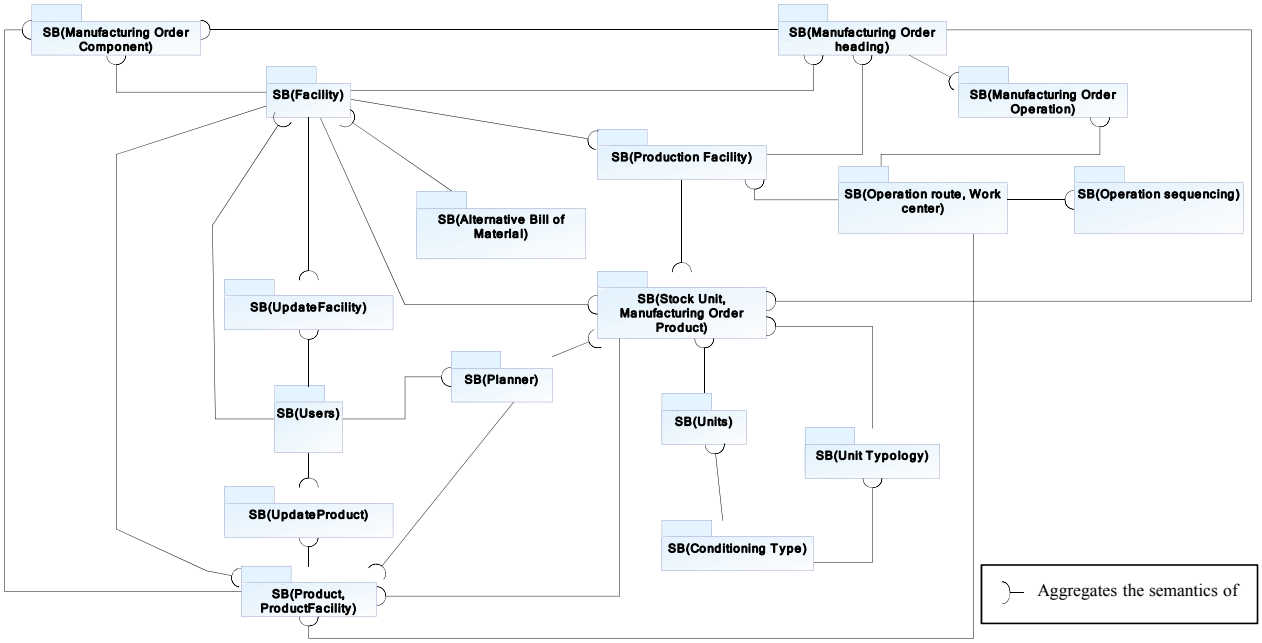


Figure 2 - The computed structured semantic model associated to the ERP Sage X3 conceptual model

There is a growing demand for integrating cooperative information systems tightly with organizational work so that these information systems can be fully, directly and immediately exploited by the intra and inter-enterprise processes (Izza, 2009). 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 for which it has been produced (conceptual and organisational interoperability).

According to (Euzenat, 2001), the encoding, lexical and syntactic interoperability levels are the most effective solutions for removing technical barriers for interoperability, but they are not sufficient to achieve a practical interoperability between computerised systems. Enabling a seamless data and model exchange at the semantic level is still a big challenge that needs a conceptual representation of the intended exchanged information and the definition of its pragmatic meaning in the context of the source and destination applications.

Interoperability problems have been classified (Panetto, 2007) with regards to a multidimensional perspective emphasising the strong relationship between models abstraction and semantics. Indeed, conceptualisation of models and its semantics extraction are still an important issue related to contextual understanding of tacit knowledge (thus hidden) embedded into those models. Our work is not only conceptualising EIS models but also it concerns also making assumptions on the mental models of the information systems' designers. The main prerequisite for achieving interoperability of information systems is then to maximise the amount of explicit semantics in the represented models (Obrst, 2003).

3 - Semantics enactment in conceptual models

As previously discussed, we focus our interest on the semantic level of interoperability and on enabling different information systems to share and use knowledge models which they represent. In order to make this possible, we consider in (Lezoche et al., 2012) two steps that need to be taken: first, we need to understand the conceptual relationships between those models in the context of their use; and second, we need to unhide the tacit knowledge buried inside them, by applying a conceptualisation approach (Guarino, 1998). 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 cannot be visualized in mind. They can be developed in different levels of abstraction of a single domain (Zdravković et al., 2011). Many conceptualisation approaches have been developed in different knowledge domains (LaOnsgri, 2009). According to (Engelbart, 1962) and (Genesereth and Nilson, 1987), developing conceptual models means specifying the essential objects, or components, of the system to be studied, the relationships of the objects that are recognised, the types of changes in the objects or their relationships which affect the functioning of the system and the types of impact these changes have on the system. These arguments are partially taken into account in our work by interpreting the semantics of the cardinality of relationships and existential constraints (mandatory elements).

Our contribution is to have at our disposal an approach that enables us fragmenting knowledge through the transformation of attributes into entities and relationships. In the proposed approach, presented in Figure 3, different inputs can be used, such as any application model, a data model, or a logical view.

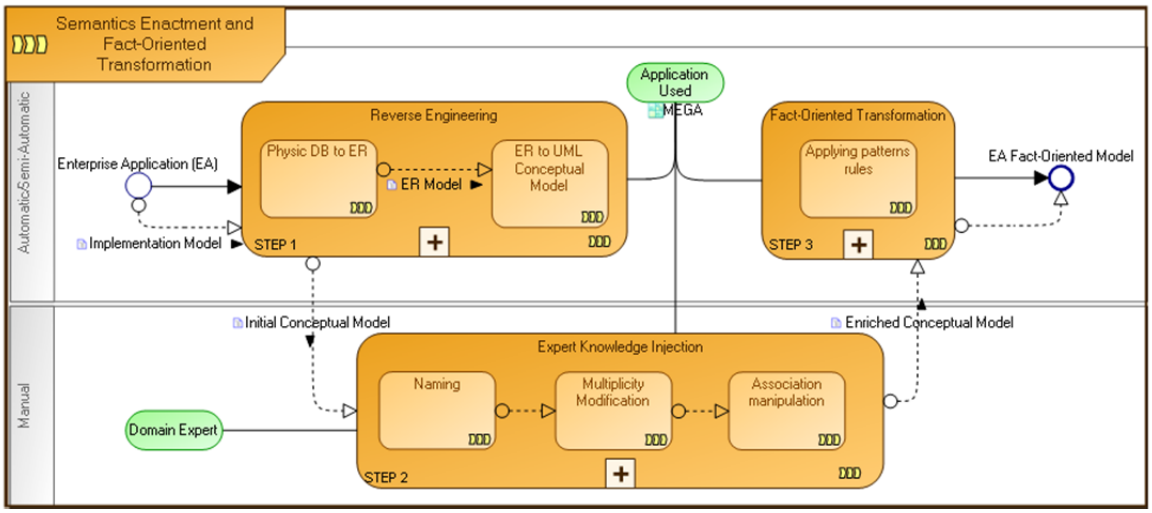


Figure 3 - Conceptualisation process (Lezoche et al, 2012)

In this approach, the initial process (Step 1) is the application of reverse engineering methods, such as (Fonkam and Gray, 1992) or (Chiang et al, 1994). However, they deliver models without any explicitation of the tacit semantics. The ADM (Architecture-Driven Modernization) initiative (OMG, 2003) from OMG (Bézivin and Kurtev, 2005) is tackling this problem by implementing a common Knowledge Discovery Meta-model to facilitate discovery of the tacit knowledge embedded inside existing software. When information systems are highly generic, the application semantics is actually captured in the populated table rows. For example, in Business Process Management systems, the structure of the enterprise processes, namely activities, associated data structures (messages), compensation and error handling blocks, etc. are defined by a system user and are not expressed by the database schema. In these cases, the intervention of the domain expert in enriching the conceptual model may be useful. Some research paths are tackling this issue by providing the tools to automatically or semi-automatically discover the semantics buried into existing data patterns (Astrova, 2004). In our scenario, we are considering that the business

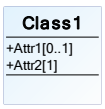
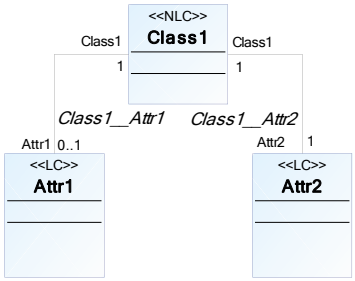
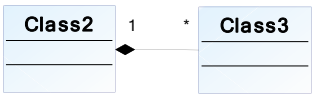
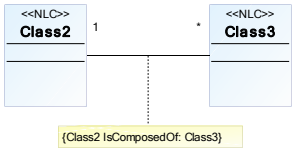
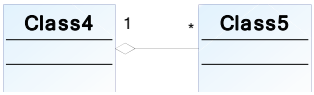
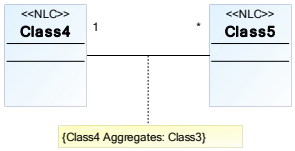
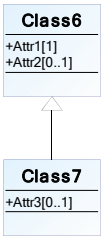
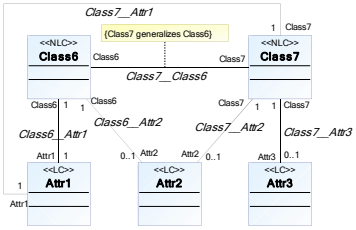
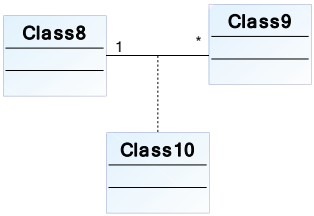
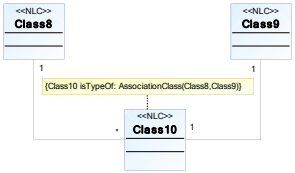
knowledge that is embedded in enterprise applications is mainly stored into the DBMS architectural level. We can then extract, from each DB, some knowledge in a form of a conceptual model, by using reverse engineering approaches. Then, knowledge extraction tools and domain experts have to enrich that model with enterprise best practices (knowledge coming from users) in the step 2. These stakeholders know the domain peculiarities and they are capable to express the specific constraints that must be embedded into the conceptual model. However, this phase must follow a structured process, in order to preserve the ontological commitment. This is particularly important when more experts are involved in the knowledge injection. In such cases, the approaches of setting up a collaborative conceptualisation processes (Guo, 2009) may be useful. Existing tools for knowledge extraction may also be used for this process (Bañares-Alcántara et al, 2003). This expert knowledge injection step is detailed in (Lezoche et al, 2012). However, the conceptual model resulting from this step is generally strongly modeller-dependent. In order to avoid semantic misinterpretations of the produced model, the step 3 aims at normalising it by fragmenting the represented knowledge into atoms of semantics. NIAM (Natural-language Information Analysis Method) (Nijssen and Halpin, 1989) 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. In the Step 3, this fact-oriented modelling (FOM) paradigm is adapted for the UML (OMG, 2004) class notation representation of the studied conceptual model. To apply this transformation, a set of modelling rules is presented on Table 1. In the resulting fact-oriented model, adding annotations preserves the semantics. The added annotations concern particular artefacts semantics such as generalisation, association class, aggregation and composite aggregation. Moreover, adapting the definitions from (Nijssen and Halpin, 1989), we define two different concept types:

- “Lexical Concept” (LC) is an object in a certain reality that can be written down. LCs always consist of letters, numbers, symbols or other characters. They can be used as

names for or references to other concepts. An attribute in UML class notation becomes a class stereotyped «LC» in FOM notation.

- "Non-Lexical Concept" (NLC), is an object in a certain reality that cannot be written down. NLCs must be named by LCs or referred to by means of LCs. A class in UML class notation becomes a class stereotyped «NLC» in FOM notation.

Table 1 - Fact-Oriented modelling patterns using UML notation (Lezoche et al, 2012)

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

Making semantics explicit

We summarise here the activities for making semantics embedded in the models explicit. After renaming the different concepts and attributes in order to be more semantically explicit, the model is updated by formalising some constraints that came after interviewing experts and users of applications. In fact, even if most of the roles constraints are automatically derived from data schemas, many of them are still hidden inside the application itself. For example, considering a particular attribute a_1, two possible redefinition cases are identified:

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

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

As for the role constraints, the majority of associations are retrieved from the model, but some implicit associations are only defined by enterprise practices and are not expressed in the model itself. The explication process applies also in this context. After this explication process, the enriched conceptual model formalises the whole application semantics (both the explicit one and the users' tacit one).

4 – A semantics structuring process

After conceptualising and enacting finest-grained semantics embedded into CISs models, resulting with a normalised FOM, a structuring process is carried out for building semantics aggregates (Yahia et al., 2012a). Each of those identified aggregates represents a "semantic molecule", composed of atomic concepts, with its own minimal mandatory semantics.

To build such aggregates, previous work (Yahia et al., 2012a) proposed a recursive approach for analysing the detailed semantics of the IS conceptual models obtained by the conceptualisation approach presented in section 3. We are considering that these models embed the whole explicit semantics of the associated IS. For the sake of self-completeness of this paper, the structuring approach is summarised in the next subsections.

4.1 Core and extended semantics

When considering an available fact-oriented conceptual model from one IS (outputs from section 3), we can distinguish the mandatory (constrained) and non-mandatory (non-

constrained) association roles, which represent mandatory and non-mandatory concepts (LCs and NLCs) expressing semantics.

The set of mandatory concepts represents all the necessary and sufficient elements that make the conceptual model semantically coherent and understandable. It comprises the non-lexical and lexical concepts linked to constrained association roles with a multiplicity equal to 1 or 1..*. On the contrary, the non-mandatory concepts correspond to the non-mandatory roles (multiplicity equal to 0..1 or *) and are only enriching the semantics of those IS conceptual models.

To some extent, the set of mandatory concepts corresponds to the core semantics that is embedded into a given IS conceptual model. The extended semantics is defined by the set of mandatory and non-mandatory concepts.

4.2 Some mathematical definitions

We define, for each IS conceptual model, the following notations.

Definition 1. C_{IS} is the set of the identified lexical and non-lexical concepts, formally defined by

$$C_{IS} = \{c_i | c_i \text{ is a lexical or a non} - \text{lexical concept from the IS conceptual model}\}$$

Moreover, we define two subsets of C_{IS} as follows:

- NLC_{IS} is the subset of C_{IS} restricted to the non-lexical concepts and,
- LC_{IS} is the subset of C_{IS} restricted to the lexical concepts.

We can note that:

$$C_{IS} = NLC_{IS} \cup LC_{IS} \text{ and } NLC_{IS} \cap LC_{IS} = \emptyset$$

Definition 2. Rel_{IS} is the set of the identified associations between concepts. Formally, it is defined by

$$Rel_{IS} = \{rel(c_i, c_j) | (c_i, c_j) \in (C_{IS})^2 \ \wedge \ c_i \ is \ associated \ to \ c_j \}$$

Definition 3. $Mult\left(rel(c_i, c_j)\right)$ is the multiplicity of the role of c_j when considering the association between c_i and c_j if it exists. For each $(c_i, c_j) \in (C_{IS})^2$, if $rel(c_i, c_j)$ exists then we have $Mult\left(rel(c_i, c_j)\right) \in \{*, 0..1, 1, 1..*\}$ and it is read c_j is associated to c_i with a multiplicity equal to $Mult\left(rel(c_i, c_j)\right)$.

Definition 4. MC_{IS} is the subset of C_{IS} restricted to mandatory concepts (the core semantics). It is formally defined by

$$MC_{IS} = \left\{c_i \middle| \exists \left(c_i, rel(c_i, c_j)\right) \in C_{IS} \times Rel_{IS} \wedge Mult\left(rel(c_i, c_j)\right) \in \{1, 1..*\}\right\}$$

Moreover, we define two subsets of MC_{IS} as follows:

- $MNLC_{IS}$ is the subset of MC_{IS} restricted to the mandatory non-lexical concepts and,
- MLC_{IS} is the subset of MC_{IS} restricted to the mandatory lexical concepts.

We can note that:

$$MC_{IS} = MNLC_{IS} \cup MLC_{IS}, \qquad MNLC_{IS} \cap MLC_{IS} = \emptyset$$

$$MNLC_{IS} = MC_{IS} \cap NLC_{IS}, \qquad MLC_{IS} = MC_{IS} \cap LC_{IS}$$

Definition 5. For each non-lexical concept c_j, we can define the set of its associated mandatory lexical concepts as follows:

$$MLC(c_j) = \left\{c_i \in LC_{IS} \middle| \left(\exists rel(c_j, c_i) \in Rel_{is} \middle| Mult\left(rel(c_j, c_i)\right) \in \{1, 1..*\}\right)\right\}$$

Definition 6. For each non-lexical concept c_j, we can define the set of its associated mandatory non-lexical concepts as follows:

$$MNLC(c_j) = \left\{ c_i \in NLC_{IS} \middle| \left(\exists rel(c_j, c_i) \in Rel_{is} \middle| Mult\left(rel(c_j, c_i)\right) \in \{1,1..*\} \right) \right\}$$

If we consider a concept defined in the context of the IS core semantics, we notice that, in order to be semantically effective in the studied domain, this concept needs to be associated on the one hand to its mandatory lexical concepts and on the other hand to other non-lexical concepts. This defines the notion of Semantic Block (SB).

4.3. Semantic blocks identification

1. *Definition*

Considering a particular non-lexical concept c_i from NLC_{IS}, a semantic block, denoted as $SB(c_i)$ and associated with the concept c_i, represents the set of the concepts necessary for the minimal semantics definition of the non-lexical concept c_i given by the conceptual model. Formally, $SB(c_i)$ is defined as follows:

$$SB(c_i) = \left\{ c_i \cup MLC(c_i) \bigcup_{c_j \in MNLC(c_i)} SB(c_j) \right\} \quad (1)$$

This definition suggests that the notion of semantic block is recursive. In the following, the meta-model of the semantic block is given and a procedure for computing all the semantic blocks of a conceptual model is proposed.

2. *Semantic block meta-model*

Here we propose to formalise the semantic block architecture through the meta-model represented on Figure 4.

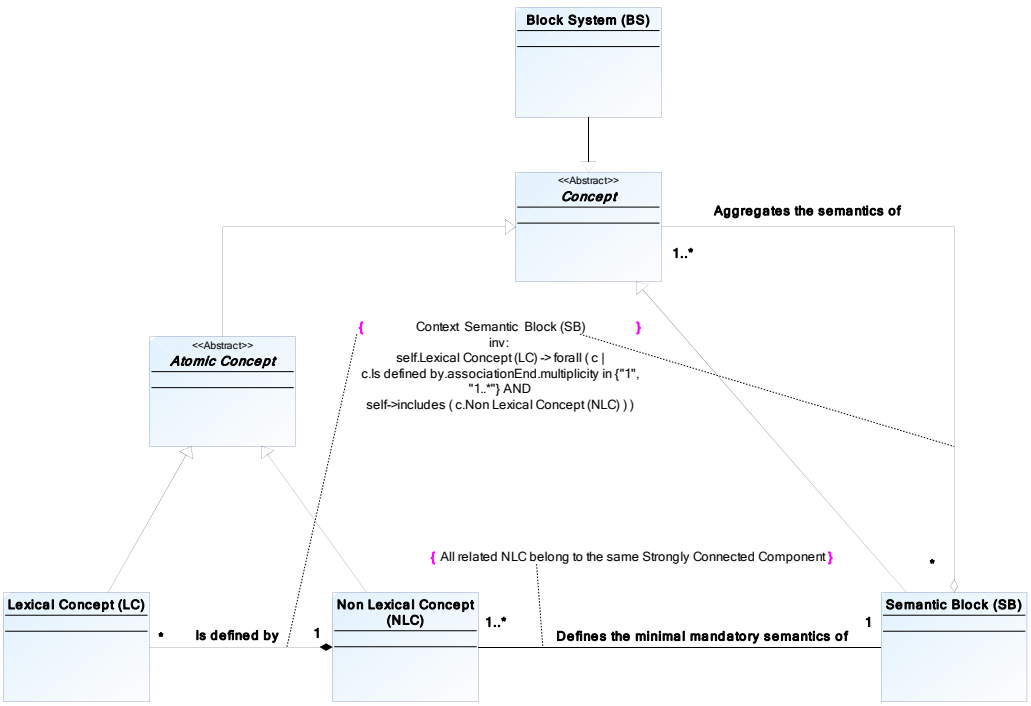


Figure 4 - Meta-model of the semantic block structure

A semantic block defines the minimal mandatory semantics of one or several non-lexical concepts such that these concepts are in the same strongly connected component[1]. Moreover, the semantics of one or several concepts can be aggregated into one or several semantic blocks. As the semantic block is a specialisation of the abstract class "Concept", its semantics can be aggregated into one or several semantic blocks of higher levels. The Block System represents the last level of aggregation and contains the minimal mandatory semantics of the studied IS conceptual model.

1. *How to build the Semantic blocks?*

Let us consider the conceptual model on Figure 5 and its transformation on Figure 6 obtained by applying the third step presented in section 3. For sake of readability, we have hidden the stereotypes («LC» and «NLC»). Let us build the semantic block of the NLC $C2$. The intrinsic

[1] A strongly connected component of a directed graph is a maximal set of vertices such that for every pair of vertices *u* and *v*, there is a directed path from *u* to *v* and a directed path from *v* to *u*.

mandatory semantics of the concept $C2$ is defined by the semantics of the mandatory lexical concepts that are associated to it, namely $A1C2$ and $A2C2$. Moreover, a given instance of the concept $C2$ exists only if it is associated to at least one instance of the concept $C5$. That means that $C5$ is mandatory for expressing the semantics of $C2$. Moreover, considering the roles of $C1$ and $C3$ in their association with $C2$, we can see that the minimal multiplicity is equal to 0. That means that the existence of any instance of $C2$ is not constrained by the existence of one instance of $C1$ or $C3$. Finally, we find again $SB(C2) = \{C2 \cup \{A1C2, A2C2\} \cup SB(C5)\}$ as in equation (1). Recursively, we can demonstrate that the intrinsic mandatory semantics of the concept $C5$ is defined by the semantics of $A1C5$ and that a given instance of the concept $C5$ exists only if it is associated to exactly one instance of the concept $C8$ and exactly one instance of the concept $C2$. That means that $SB(C5) = \{C5 \cup \{A1C5\} \cup SB(C2) \cup SB(C8)\}$.

Applying the same reasoning, we can build $SB(C8)$ as follows: $SB(C8) = \{C8 \cup \{A1C8\}\}$. Finally we can deduce that: $SB(C2) = \{\{C2, C5, C8\} \cup \{A1C2, A2C2\} \cup \{A1C5\} \cup \{A1C8\}\}$.

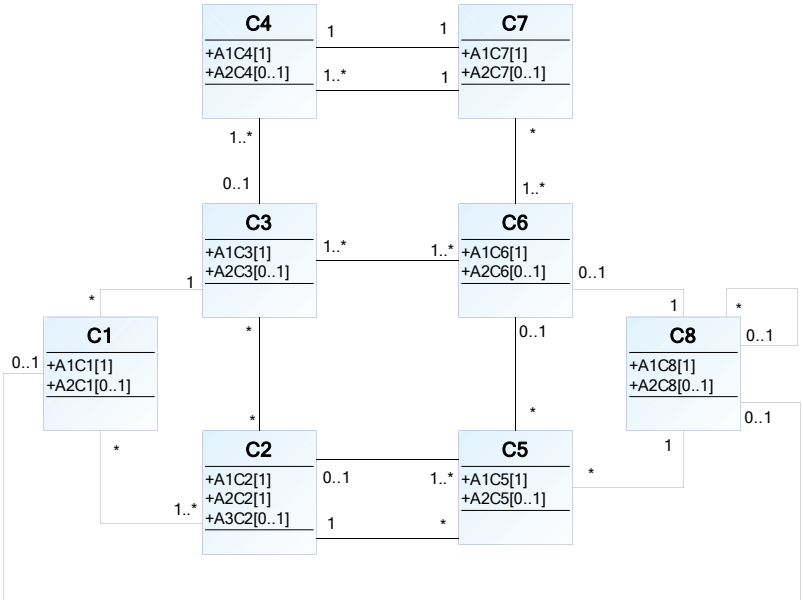


Figure 5 - An instance of conceptual model

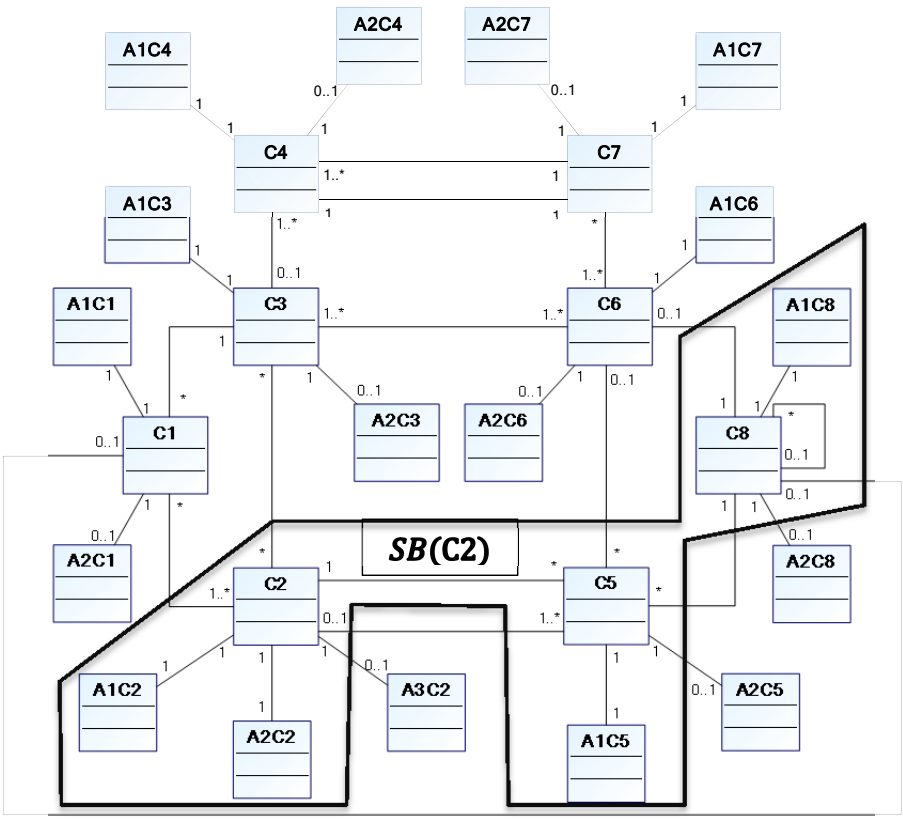


Figure 6 - "Fact-oriented modelling" transformation of the model of Figure 5

To simplify the computation of the semantic block of one concept c_i, (Yahia et al, 2012a) propose, first, to identify the set of non-lexical concepts that are included in the semantic block and, second, to add the associated mandatory lexical concepts. That means that $SB(c_i)$ is determined as follows: $SB(c_i) = SB_c(c_i) \cup SB_a(c_i)$ with

- $SB_c(c_i) = \left\{\{c_i\} \cup_{c_j \in MNLC(c_i)} SB_c(c_j)\right\}$ and,
- $SB_a(c_i) = \{MLC(c_j) | c_j \in SB_c(c_i)\}$

For instance, $SB_c(C2) = \{C2, C5, C8\}$ and $SB_a(C2) = \{A1C2, A2C2, A1C5, A1C8\}$.

4.5 A procedure to compute the semantic blocks

(Yahia et al, 2012a) propose the following procedure to compute all the semantic blocks of a given conceptual model:

i. Building the associated semantic-dependency graph.

ii. Building the graph of the strongly connected components based on the semantic-dependency graph.

iii. Computing the semantic blocks SB_c associated with each strongly connected component.

iv. Computing, for each SB_c, the semantic block SB_a by adding all the mandatory lexical concepts associated to each non-lexical concept from SB_c.

v. Computing $SB = SB_c \cup SB_a$.

These steps are summarised as follows.

4.5.1 Building the associated semantic-dependency graph

To facilitate the building of the semantic blocks for each c_i from NLC_{IS}, the associated set $SB_c(c_i)$ are identified by using graph theory modelling and its associated mathematical tools. The conceptual model can be associated with what we call a semantic-dependency graph. Each node represents a non-lexical concept of the conceptual model. Each edge is built from the conceptual model as follows: the edge (c_i, c_j) exists if (i) there is an association between c_i and c_j in the conceptual model, and (ii) if the minimal multiplicity for the role of c_j is equal to 1 $\left(c_j \in MNLC(c_i)\right)$. That means that the existence of the edge (c_i, c_j) represents the fact that c_j is mandatory for expressing the semantics of c_i. The Figure 7 shows the semantic-dependency graph associated with the conceptual model of the Figure 5.

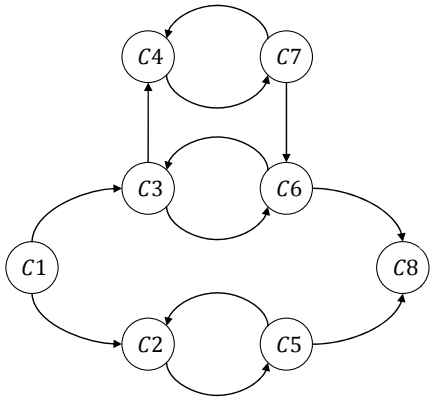


Figure 7 - Semantic-dependency graph associated with the conceptual model of Figure 5

4.5.2 Building the graph of the strongly connected components

(Yahia et al., 2012a) demonstrate that there is one semantic block per strongly connected component of the semantic-dependency graph. That means that for building the semantic blocks, only one concept in a given strongly connected component are considered (the other concepts share the same semantic block). The semantic-dependency graph is thus simplified by considering an equivalent graph where the nodes represent each strongly connected component of the former semantic-dependency graph. On this graph, one of these nodes (e.g. SCCi) is connected to another node (e.g. SCCj) if there exists at least one edge from a concept from SCCi to a concept from SCCj. Identifying all the strongly connected components of a graph is a well-known problem in graph theory that can be solved with polynomial effort by using Kosaraju-Sharir's algorithm (Sharir, 1981). The graph of the strongly connected components related to the semantic-dependency graph of Figure 7 is given on Figure 8. On this graph, the strongly connected components are defined as follows: $\mathrm{SCC1} = \{\mathrm{C1}\}$, $\mathrm{SCC2} = \{\mathrm{C2}, \mathrm{C5}\}$, $\mathrm{SCC3} = \{\mathrm{C3}, \mathrm{C4}, \mathrm{C6}, \mathrm{C7}\}$ and $\mathrm{SCC4} = \{\mathrm{C8}\}$.

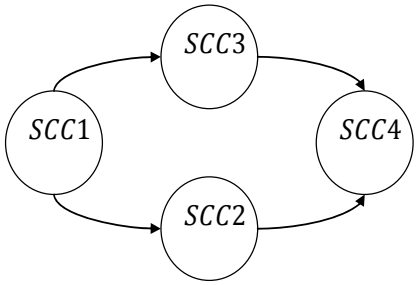


Figure 8 - Graph of the strongly connected components related to the graph of Figure 7

4.5.3 Computing SB_c associated with each strongly connected component

(Yahia et al., 2012a) propose two algorithms for automatically computing $\mathrm{SB_c}$. These algorithms are based on paths discovery in the graph. In fact, if there exists one path from the node $SCCi$ to the node $SCCj$ then $SCCj$ is included into $SB_c(SCC\mathrm{i})$. By applying these algorithms, the following semantic blocks are obtained:

- $SB_c(SCC1) = SCC1 \cup SCC2 \cup SCC3 \cup SCC4$,
- $SB_c(SCC2) = SCC2 \cup SCC4$,
- $SB_c(SCC3) = SCC3 \cup SCC4$ and
- $SB_c(SCC4) = SCC4$.

And finally, replacing the strongly connected components by their content, the following semantic blocks are obtained:

- $SB_c(C1) = \{C1, C2, C3, C4, C5, C6, C7, C8\}$,
- $SB_c(C2, C5) = \{C2, C5, C8\}$,
- $SB_c(C3, C4, C6, C7) = \{C3, C4, C6, C7, C8\}$ and
- $SB_c(C8) = \{C8\}$.

4.5.4 Computing, for each SB_c, the semantic block SB_a

Each semantic block SB_a contains the mandatory lexical concepts associated to the non-lexical concepts in SB_c. By applying the definition of SB_a $\left(SB_a(c_i) = \{MLC(c_j) | c_j \in SB_c(c_i)\}\right)$ on the instance of Figure 6 we obtain:

- $SB_a(\mathrm{C1}) = \{A1C1, A1C2, A2C2, A1C3, A1C4, A1C5, A1C6, A1C7, A1\mathrm{C8}\}$,
- $SB_a(\mathrm{C2}, \mathrm{C5}) = \{A1C2, A2C2, A1C5, A1\mathrm{C8}\}$,
- $SB_c(\mathrm{C3}, \mathrm{C4}, \mathrm{C6}, \mathrm{C7}) = \{A1C3, A1C4, A1C6, A1C7, A1\mathrm{C8}\}$ and
- $SB_c(\mathrm{C8}) = \{A1\mathrm{C8}\}$.

4.5.5 Computing each semantic block SB

Each semantic block SB is the union of SB_c and SB_a. By applying this definition on the instance of Figure 6 we obtain:

- $SB(C1) =$
  $\{\{C1, C2, C3, C4, C5, C6, C7, C8\} \cup$
  $\{A1C1, A1C2, A2C2, A1C3, A1C4, A1C5, A1C6, A1C7, A1C8\}\}$,
- $SB(\mathrm{C2}, \mathrm{C5}) = \{\{C2, C5, C8\} \cup \{A1C2, A2C2, A1C5, A1\mathrm{C8}\}\}$,
- $SB(\mathrm{C3}, \mathrm{C4}, \mathrm{C6}, \mathrm{C7}) = \{\{C3, C4, C6, C7, C8\} \cup \{A1C3, A1C4, A1C6, A1C7, A1\mathrm{C8}\}\}$ and
- $SB(\mathrm{C8}) = \{C8\} \cup \{A1\mathrm{C8}\}$.

For validating our approach, next section will detail an industrial case study involving two enterprise information systems that need to interoperate: Sage X3 ERP and Flexnet MES. We will also present the prototype of the tool that implements the whole approach for automatizing the model conceptualisation and semantics structuring processes.

5 - Case study

Interoperability between organisational and manufacturing activities is crucial in manufacturing enterprises. Production services have to produce, quickly and efficiently, the right volume of the right product at the right moment. For this reason, they need real time information coming from others services, which need in return a precise and updated data from the production. We propose here to study and present the first part of such a Business to Manufacturing (B2M) interoperability issue by studying Sage X3 as an Enterprise Resource Planning (ERP) application and Flexnet as a Manufacturing Execution System (MES). In order to illustrate our approach, we will detail the conceptualisation process applied to a subset of the ERP information system model in section 5.1. Section 5.2 will detail the semantics structuration process (computing semantic blocks) applied to the same subset. Section 5.3 will present the results coming from Flexnet MES application. Section 5.4 will discuss about the contributions drawing some limits. For sake of readability, in the following, we will name each subset of models by the name of the related enterprise applications.

5.1 Conceptualisation of Sage X3 ERP model

The objectives of this case study are (i) to analyse how the manufacturing order process inside the ERP Sage X3 application is modelled, (ii) to use the proposed modelling process for making the implicit knowledge explicit in the model structure.

The model depicted on Figure 9 is the output from the two first steps proposed in (Lezoche et al., 2012). This means that we have already completed the “Reverse Engineering” and the “Expert knowledge injection” steps. The “Manufacturing Order heading” concept represents the management function of production orders and planned activities. This function allows the generation of a manufacturing order by variation of one or more classifications and a single production line. For each manufacturing order, the achievement of the material benefits and sequencing operations is possible. The function captures general information, such as

planning and production facility and the status of the order. It allows entering general information about the production order. The availability of components is then checked through the information given by the bill of material related to the launched products. Once the above initial information is determined, the system updates the list of materials and operations of the created or modified orders.

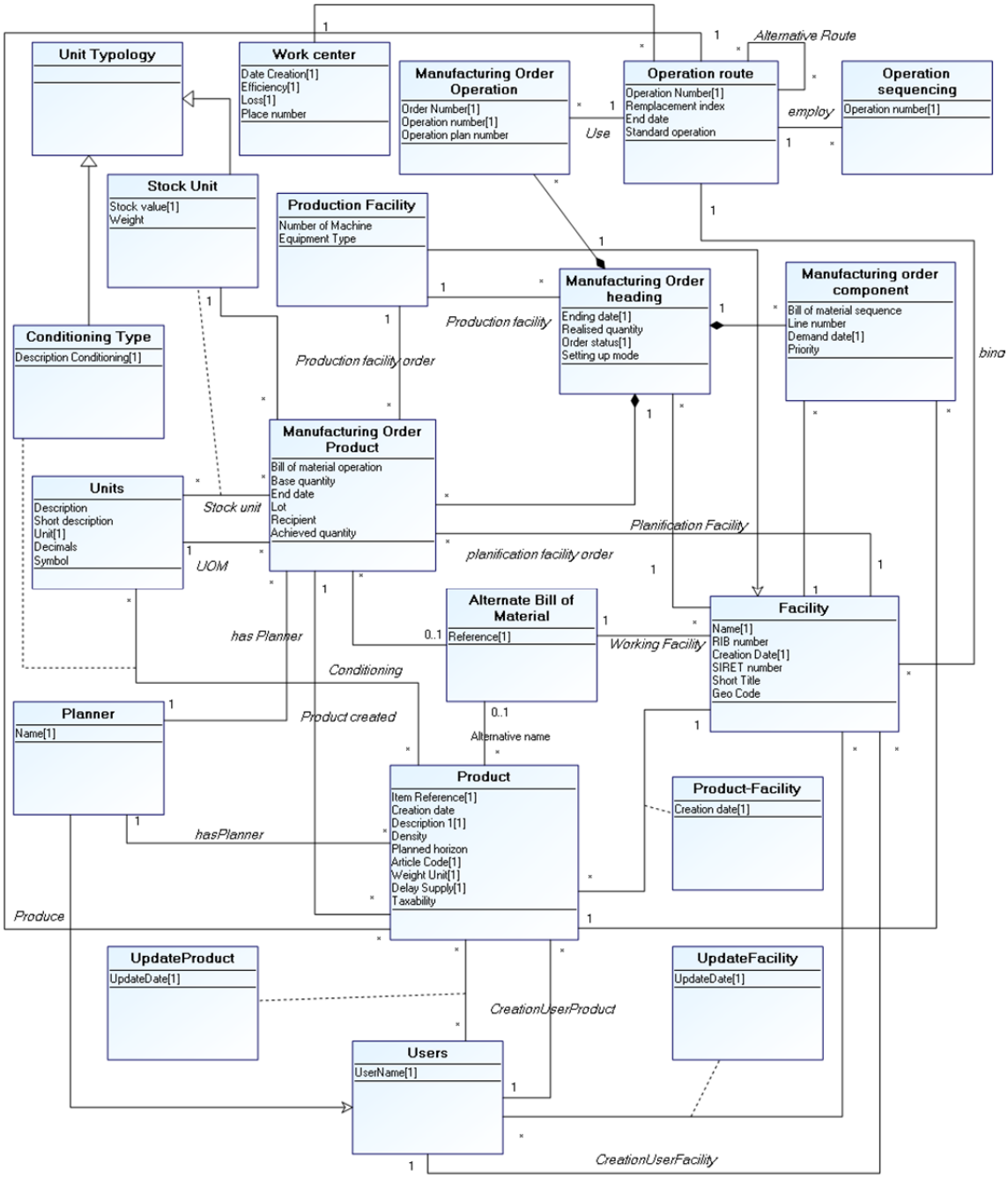


Figure 9 - Enriched Sage X3 manufacturing order process model

We then applied the pattern transformation rules, presented in the previous section, to increase the granularity of the knowledge embedded into the model. Figure 1 shows the resulting FOM after applying our approach to the Sage X3 manufacturing order process. It presents the explicit and mandatory knowledge linked to the Sage X3 process and it contains the semantic annotations that preserve relationships information for assuring the traceability of the sustainability receipts. As can be evinced by the model, in the Figure 1, there are some pivotal concepts in this process like “Product”, “Facility”, “Manufacturing Order Product”, “Manufacturing Order heading” and “Operation route” that concentrate the majority of the relationships and that transmit the core semantics of the described process. At the first glance, it seems that the resulting model is much more complex than the initial one. This may looks true from a visual point of view, but it is false in terms of simplicity to treat the model’s semantics. Indeed, the atomic semantics is made explicit, which helps any automatic computing. An important result is that using the model with such high level of granularity will facilitate automatic computing for semantic gap evaluation.

5.2 Semantics structuring of SageX3 ERP model

The reminding of this section will apply our semantics structuring approach to build semantic-dependency graphs for automatically identifying aggregates of semantics (so called semantics blocs) (Figure 2). This simplified model facilitates the identification of semantically self-contained subsystems, and further interoperation analysis, knowledge discovery capabilities and systems adaptations for enhancing sustainability.

1) Building the associated semantic-dependency graph

As presented in section 4.5 the semantic-dependency graph, related to the conceptual model of SageX3, is built by considering each association between two concepts and then building an edge if the minimal multiplicity for the role is equal to 1. The result of this procedure is presented on Figure 10.

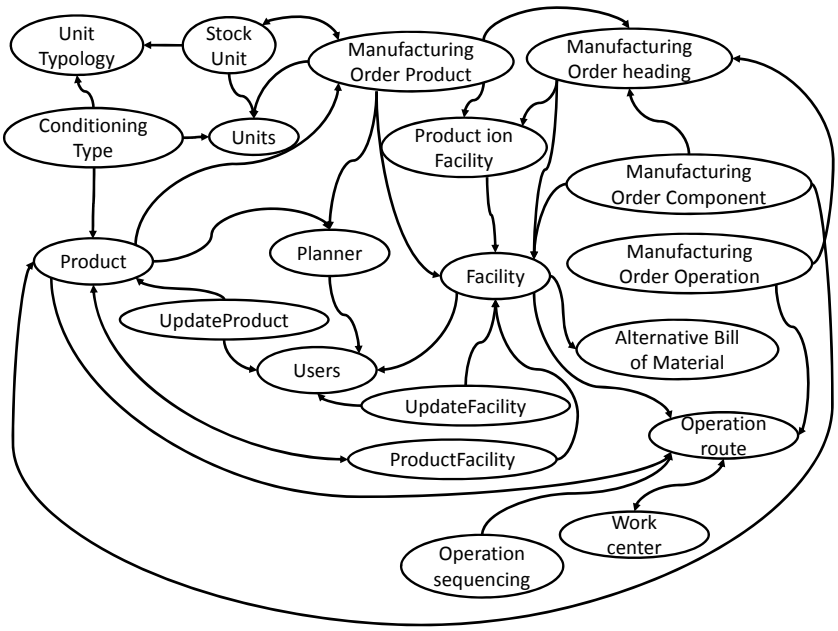


Figure 10 - Semantic-dependency graph related to the conceptual model of SageX3 ERP

2) Building the graph of the strongly connected components (SCC) based on the semantic-dependency graph

The subsequent step is the building of the graph of the strongly connected components (Figure 11), related to the semantic-dependency graph of SageX3 ERP (Figure 10). We can note that three merged nodes has been built (namely SCC1, SCC2, SCC3) representing three strongly connected components. $SCC_1 = \{Stock\ Unit; Manufacturing\ Order\ Product\}$, $SCC_2 = \{Product; ProductFacility\}$ and $SCC_3 = \{Operation\ route; Work\ center\}$. All the other strongly connected components consist of only one concept.

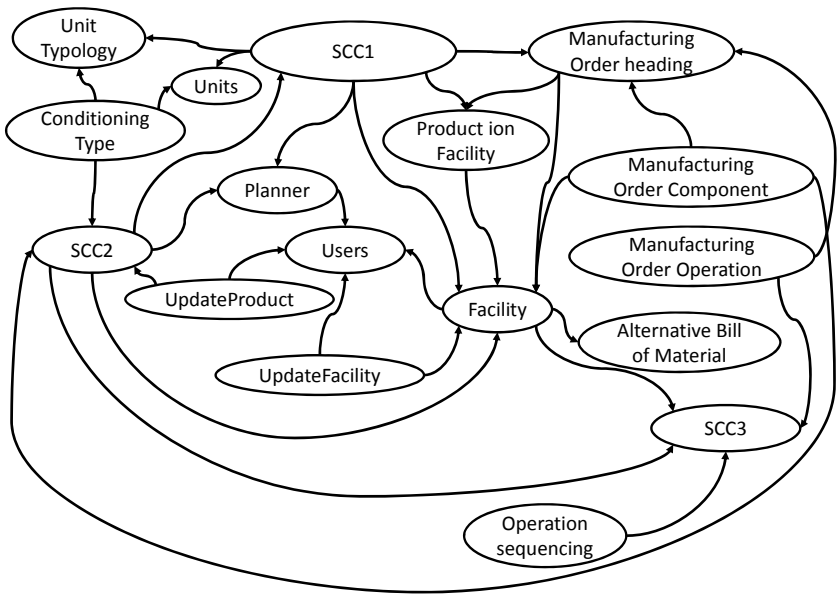


Figure 11 - Graph of the SCCs related to the semantic-dependency graph of SageX3 ERP

Figure 2 provides a model representing all the semantic blocks related to the Sage X3 manufacturing order process model and their semantics aggregation relationships.

5.3 Conceptualising and structuring semantics of Flexnet MES model

Applying the same procedures, we first obtain the enriched fact-oriented model of the purchase order process in Flexnet application (Figure 12). Then implemented tool computes the semantic blocks as shown on Figure 13.

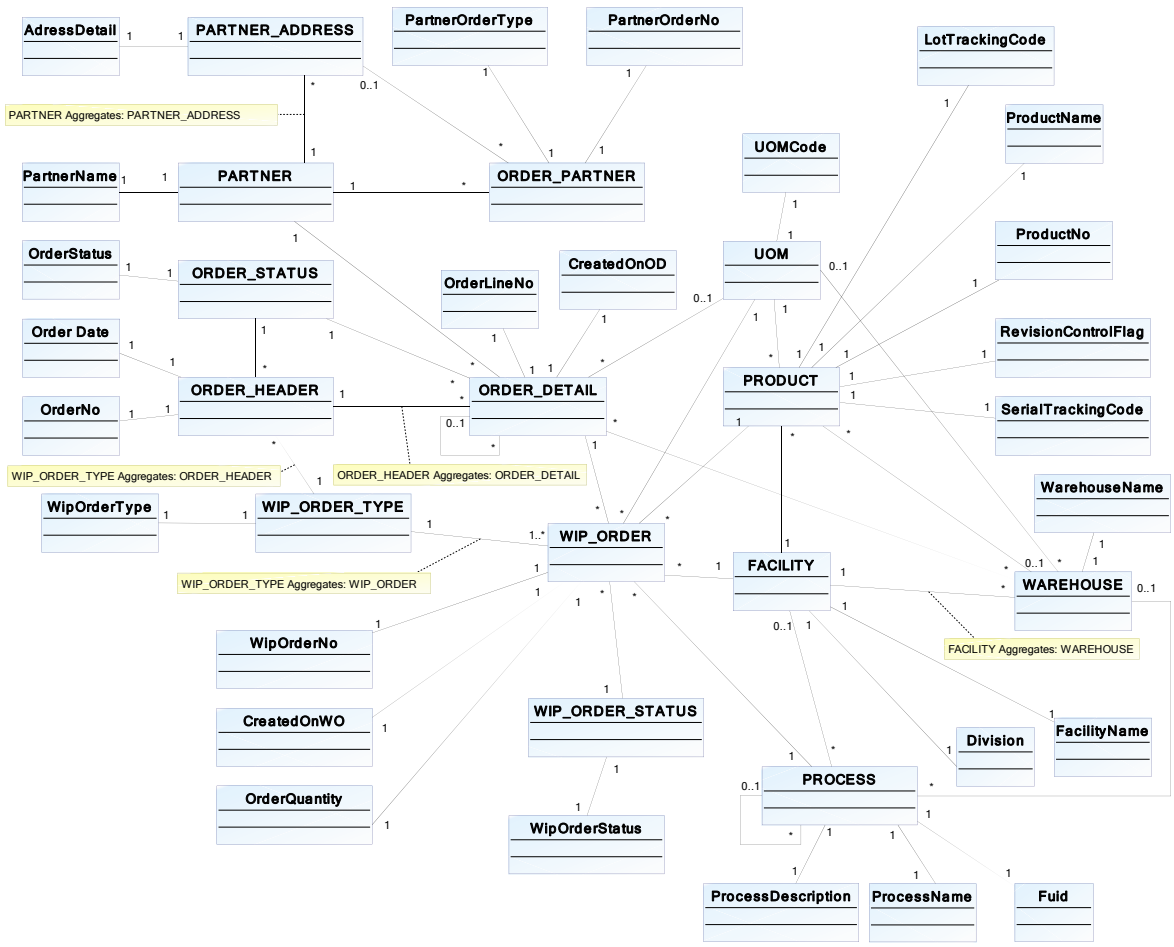


Figure 12 - Enriched fact-oriented model of the purchase order process in Flexnet application

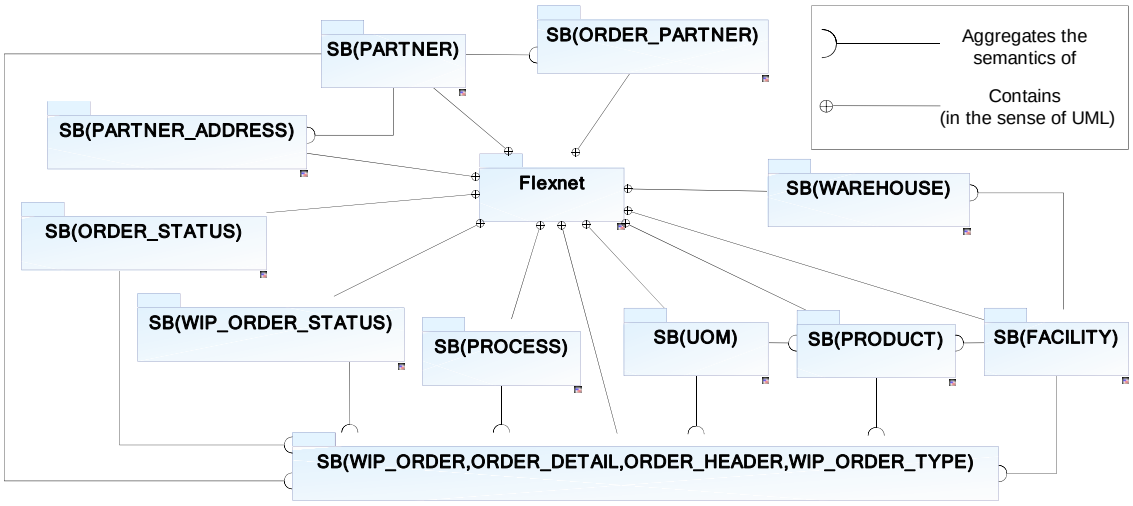


Figure 13 - The computed semantic blocks related to Flexnet MES

The different procedures presented in this paper have been implemented in the MEGA Suite environment[2]. MEGA Suite supports UML notations and allows building our own meta-model based on its ad-hoc MOF[3] meta-model. The meta-model presented on Figure 4 has been implemented. In this implementation, the semantic block is conceptualised as a UML package and concepts are conceptualised as UML classes stereotyped as «LC» or «NLC». The implemented procedures assist the expert in the conceptualisation process (step 2) and fully automatized the generation of the FOM model and the identification of all the semantic blocks and their relationships.

5.4 Discussion

Applying the whole process for conceptualising and structuring the semantics to the Sage X3 and Flexnet applications, we obtain, for each application:

- one normalised conceptual model that embeds the whole semantics (coming directly from the data schema and enriched by expert knowledge injection) and
- one structuration of this semantics through the semantic blocks that emphasizes the dependencies between concepts and that highlights independent semantics aggregates.

These simplified semantic models allow a better understanding of each application's functionalities. An engineer can then focus on specific semantically independent components (i.e. the semantic blocks) of these applications when analysing their interoperation capabilities. When defining what information has to be exchanged, the semantic block allows defining priorities related to the complexity of the model or to some initial requirements. As an example, when dealing with manufacturing orders in Sage X3 application, the semantic block $SB(Stock\ Unit; Manufacturing\ Order\ Product)$ contains all the concepts that are mandatory for Sage X3 to operate correctly and thus defines the interoperability requirements when such information is picked up from another application.

[2] MEGA international, http://www.mega.com
[3] OMG's MetaObject Facility: http://www.omg.org/mof/

Those models are normalised, that means that they do not contain any modelling patterns introduced by some modellers' practices. Thus, they can be directly used by some engineers or automatized tools without any ambiguity in their interpretation.

The work presented in this paper contributes to new research dimensions related to sustainable interoperability by facilitating knowledge conceptualisation and structuration into a set of discovered semantic blocks, thus improving learning capacity to better take into account changes when adaptation is required. Moreover, this structured model allows deeply analysing interoperability requirements and thus specifying in which way a network node should react in order to enable the system, as well as the entire network, to evolve for the better sustainable interoperable state.

However, we can draw some limits of our approach. One major assumption is necessary for the approach to be practical: the application must expose its implemented data schema and experts are assumed to be able to update the conceptual model with the many constraints and practices that are not implicitly expressed in the schema. Moreover, the size of a real application schema is so huge (more than 800 tables and 600 relationships for Sage X3) that this semi-automatically implemented approach remains complex to apply due to the necessary implication of one expert.

6 Conclusions

In this paper, we proposed a conceptualisation approach for enacting implicit semantics embedded into Enterprise Information System models by a deep analysis of existing data models enriched by users' and experts' knowledge. This approach is composed of 3 steps, staging from the traditional database reverse engineering process, through a knowledge elicitation and model enrichment by domain experts, to the application of formal fact-oriented modelling rules for externalising tacit semantics. Moreover, in order to structure the whole

semantics into independent aggregates that may emphasize subsystems, we defined the concept of semantic block (SB) and we developed an automatized procedure to compute these SBs. The resulting semantics architecture allows the identification of semantically self-contained subsystems, facilitating further interoperation analysis, knowledge discovery capabilities and systems adaptations for enhancing sustainability. The conceptualisation and structuring processes have been validated on a case study involving two industrial Enterprise Applications, demonstrating the applicability of our approach. This approach contributes to the sustainability aspects of the information systems management. In fact, it aims at extracting, formalizing and using the tacit knowledge for facilitating the maintenance of the interoperability between enterprises when major changes occur.

Further works concern using the resulting semantic conceptual model and architecture for facilitating the assessment of the (non)-interoperation barriers between Enterprise Information Systems or some of their subsystems (identified, for instance, by the semantic blocks). The resulting analysis, based on an interoperability measures map, can help information technology consulting companies for parameterising and integrating enterprise applications (ERP, MES…) taking into account interoperability constraints.

References

Astrova, I., 2004. Reverse Engineering of Relational Databases to Ontologies. In *Lecture Notes in Computer Science*, LNCS 3053, pp. 327-341, Springer.

Bañares-Alcántara R., Aldea, A., Bocio, J., Isern, D., Kokossis, A., 2003. Deliverable 5 – Knowledge Discovery Tools, h-TechSight project, IST-2001-33174.

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

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*. Volume 12, Number 2, pp. 107-142.

EIF, 2010. European Interoperability Framework for European Public Services Version 2.0, IDABC, European Commission. http://ec.europa.eu/isa/documents/isa_annex_ii_eif_en.pdf. Accessed on 2011-12.

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

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.

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.

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

Genesereth, M.R., Nilsson, N.J., Logical Foundation of Artificial Intelligence, Morgan Kauffman, Los Altos, CA, 1987

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

Guo, J., 2009. Collaborative conceptualisation: towards a conceptual foundation of interoperable electronic product catalogue system design. Enterprise Information Systems, 3 (1) 59-94.

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

Izza, S., 2009. Integration of industrial information systems: from syntactic to semantic integration approaches. *Enterprise Information Systems*. 3(1), pp. 1-57, Taylor & Francis.

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

Lezoche, M, Panetto, H., Aubry, A., 2012. Formal Fact-Oriented model transformations for Cooperative Information Systems semantic conceptualisation. Enterprise Information Systems. *Lecture Notes in Business Information Processing.* LNBIP 102, pp. 117-131, Springer.

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

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

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

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

Panetto, H., 2007. Towards a Classification Framework for Interoperability of Enterprise Applications. *International Journal of CIM*, Taylor & Francis, 20 (8), 727-740, ISSN: 0951-192X

Sharir, M., 1981. A strong-connectivity algorithm and its applications in data flow analysis. In *Computers and Mathematics with Applications*. Volume 7, pp. 67-72.

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.

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

Yahia, E., Lezoche, M., Aubry, A., Panetto, H, 2012a. Semantics enactment for interoperability assessment in Enterprise Information Systems. *Annual Reviews in Control,* 36(1), pp. 101-117.

Yahia E., Aubry A., Panetto H., 2012b. Formal measures for semantic interoperability assessment in cooperative enterprise information systems. *Computers in Industry*, 63 (5), pp. 443-457.

Zdravković. M., Panetto, H., Trajanović, M., Aubry, A., 2011, An approach for formalising the supply chain operations, Enterprise Information Systems, 5 (4) 401-421, doi:10.1080/17517575.2011.593104.