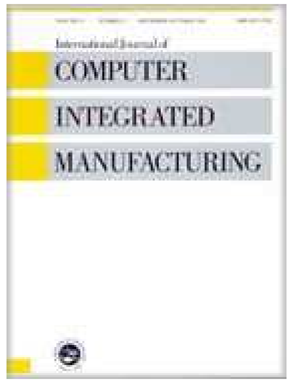
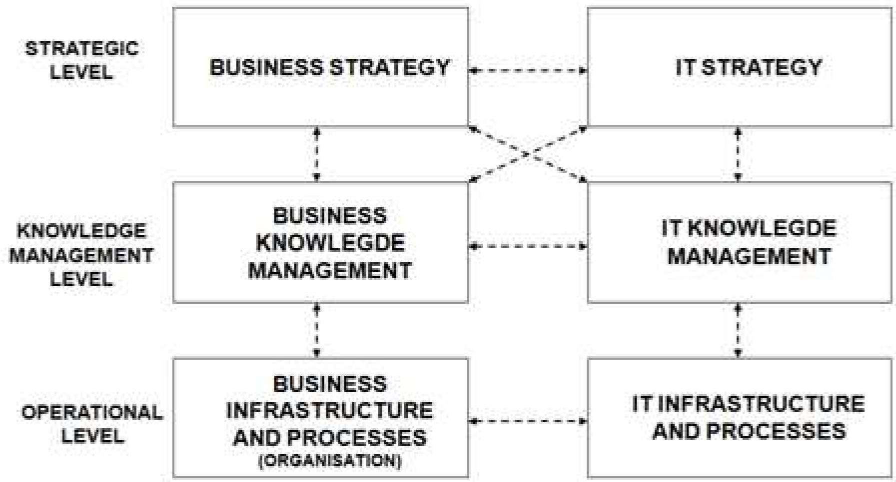
An ontological approach for strategic alignment: a SCOR case study

Omar Sakka, Pierre-Alain Millet, Valérie Botta-Genoulaz



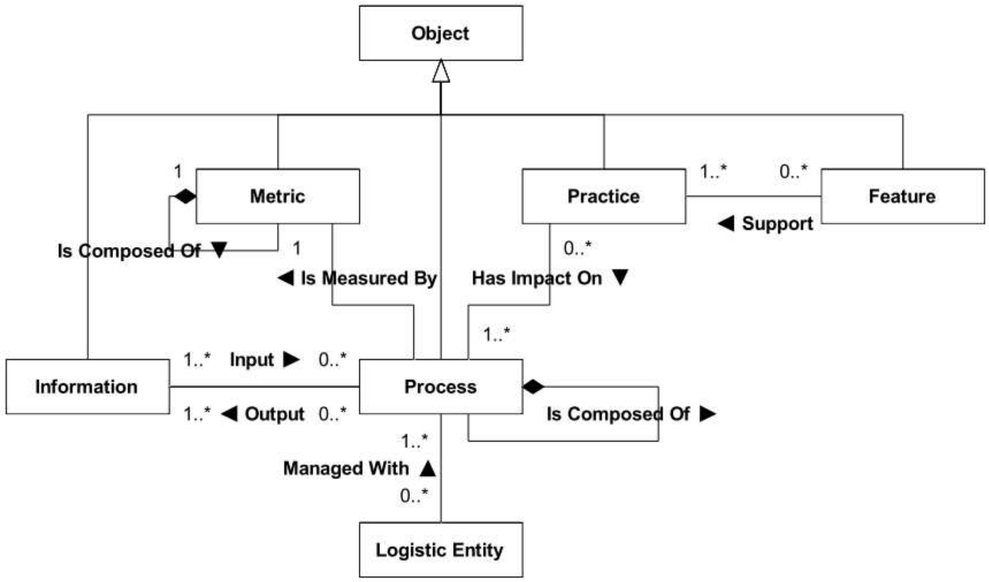
(Enterprise) An ontological approach for strategic alignment: a SCOR case study

Journal:	*International Journal of Computer Integrated Manufacturing*
Manuscript ID:	TCIM-2011-IJCIM-0021.R1
Manuscript Type:	Special Issue Paper
Date Submitted by the Author:	18-Mar-2011
Complete List of Authors:	Sakka, Omar; INSA Lyon, LIESP, Computer Engineering MILLET, Pierre-Alain; INSA Lyon, LIESP, Computer Engineering Botta-Genoulaz, Valérie; INSA Lyon, LIESP, Industrial Engineering
Keywords:	BUSINESS PROCESS RE-ENGINEERING, ENTERPRISE MODELLING
Keywords (user):	Strategic alignment, OWL



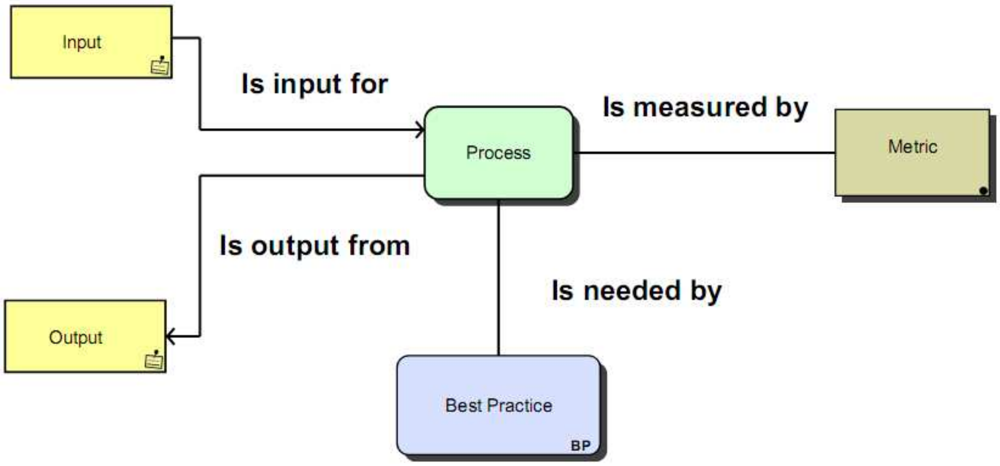
Strategic Alignment Model complemented with knowledge management components (Gudas, Saulius et al., 2006)
119x66mm (96 x 96 DPI)

SCOR Model - Supply Chain Council (SCC, 2009)
322x121mm (96 x 96 DPI)

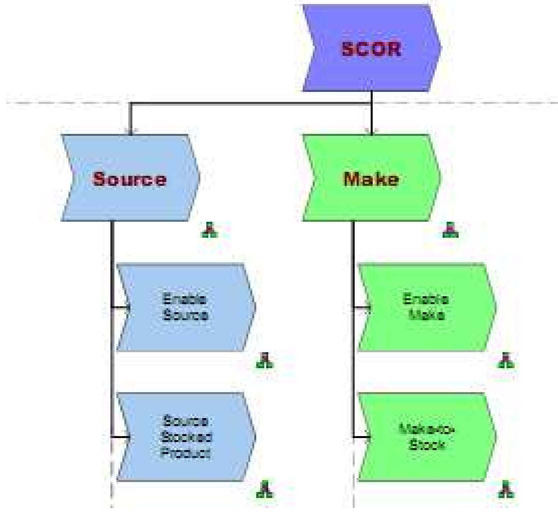


SCOR meta-model (Millet and Botta-Genoulaz, 2006)
229x134mm (96 x 96 DPI)

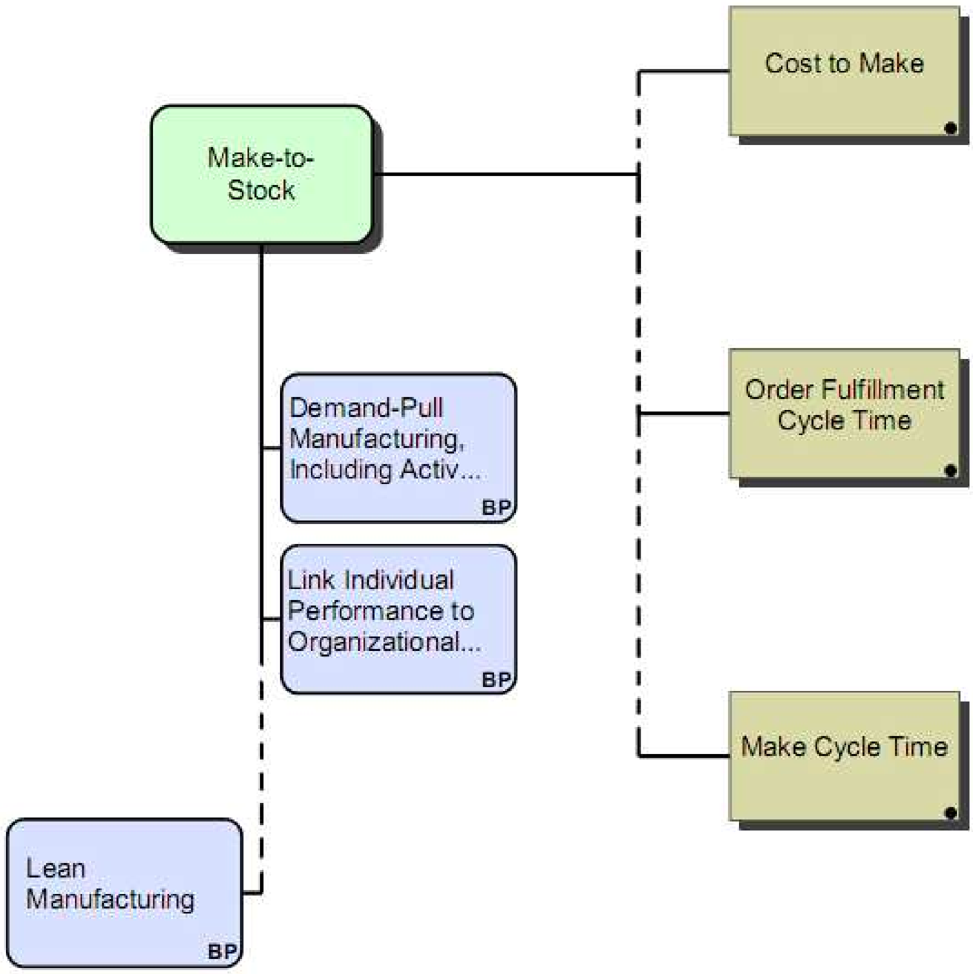
Tree of the SCOR/ARIS model
57x40mm (96 x 96 DPI)



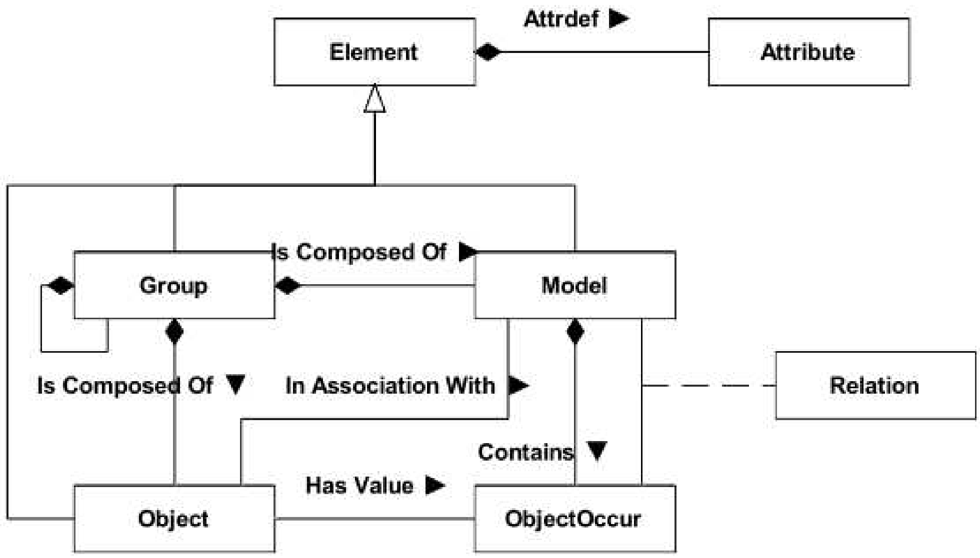
ARIS concepts supporting SCOR/ARIS
217x103mm (96 x 96 DPI)



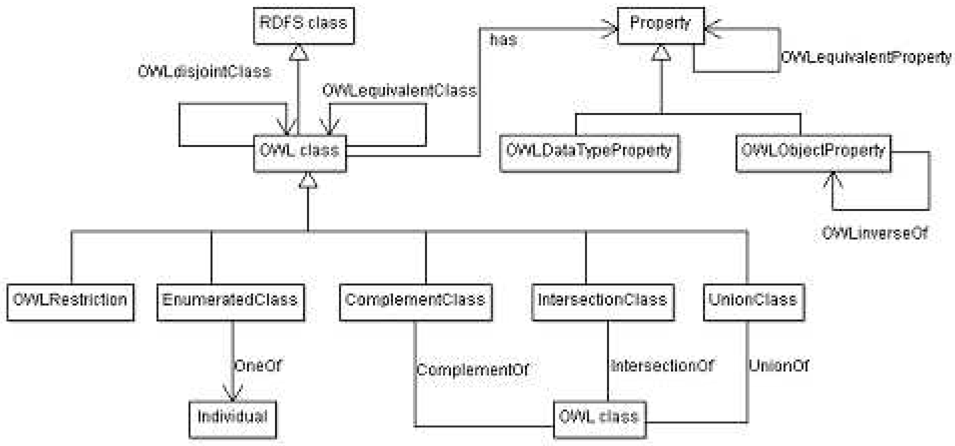
SCOR level 1 and 2 process overview
74x67mm (96 x 96 DPI)



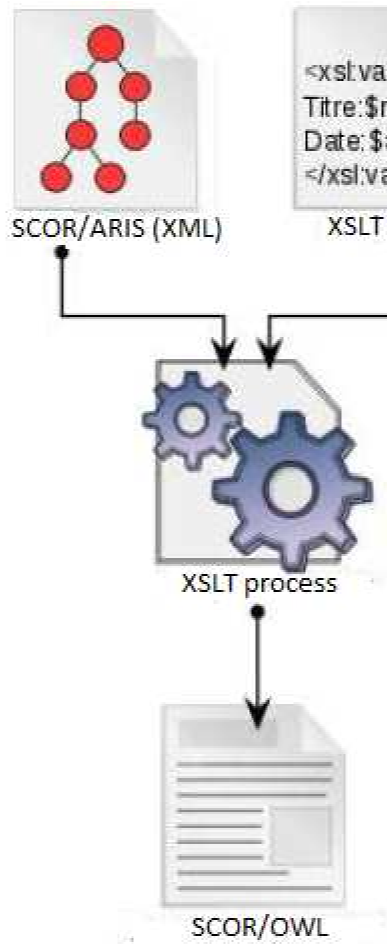
Function Allocation Diagram of the Make-to-stock process
145x144mm (96 x 96 DPI)



ARIS meta-model export
149x85mm (96 x 96 DPI)



OWL meta-model
126x62mm (96 x 96 DPI)



Transformation process
63x124mm (96 x 96 DPI)

```
<xsl:for-each select="Group/Group">
<xsl:element name="Group1">
<xsl:text>&#10;</xsl:text>
 <xsl:call-template name="concept-group"/>
<xsl:text>&#10;</xsl:text>
<xsl:element name="Group2">
<xsl:value-of select="../AttrDef/AttrValue"/>
</xsl:element>
</xsl:element>
```

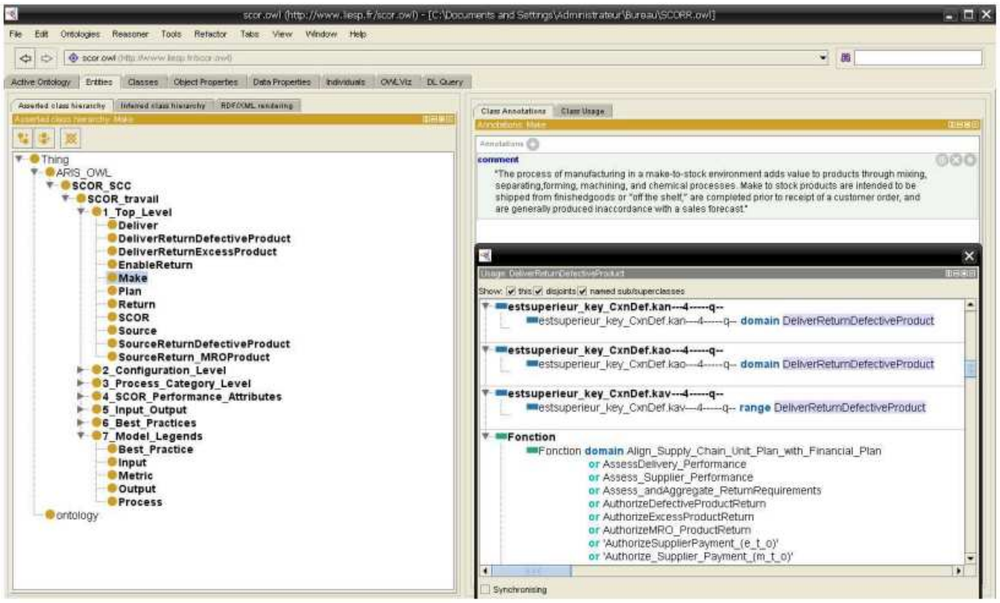
An XSLT rule
112x40mm (96 x 96 DPI)

```
<xsl:when test="@CxnDef.Type='CT_IS_NEEDED_BY'">
<xsl:attribute name="rdf:about">
<xsl:value-of select="concat('Is_Needed_By','_',
(translate(../AttrDef/AttrValue,':;,?./§%µ-£
"<&>#|°}{@~-','_______________________')))" />
</xsl:attribute>
</xsl:when>
```

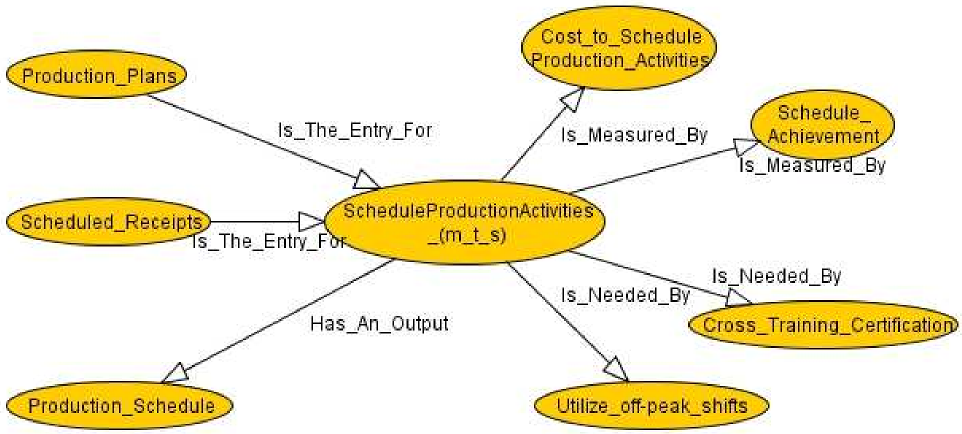
XSLT rule for “Is Needed by” relationship
112x39mm (96 x 96 DPI)

```
<xsl:for-each select="Group/ObjDef">
- <xsl:for-each select="CxnDef">
- <xsl:element name="owl:ObjectProperty">
  <xsl:call-template name="recherche2" />
- <xsl:element name="rdfs:domain">
- <xsl:attribute name="rdf:resource">
  <xsl:value-of select="concat('#' , (translate(../AttrDef/AttrValue,':;,?./§%µ-£
"<&>#|°}{@~-',' _______________________')))" />
  </xsl:attribute>
  </xsl:element>
```

XSLT rule for creating owl ObjectProperty
112x44mm (96 x 96 DPI)



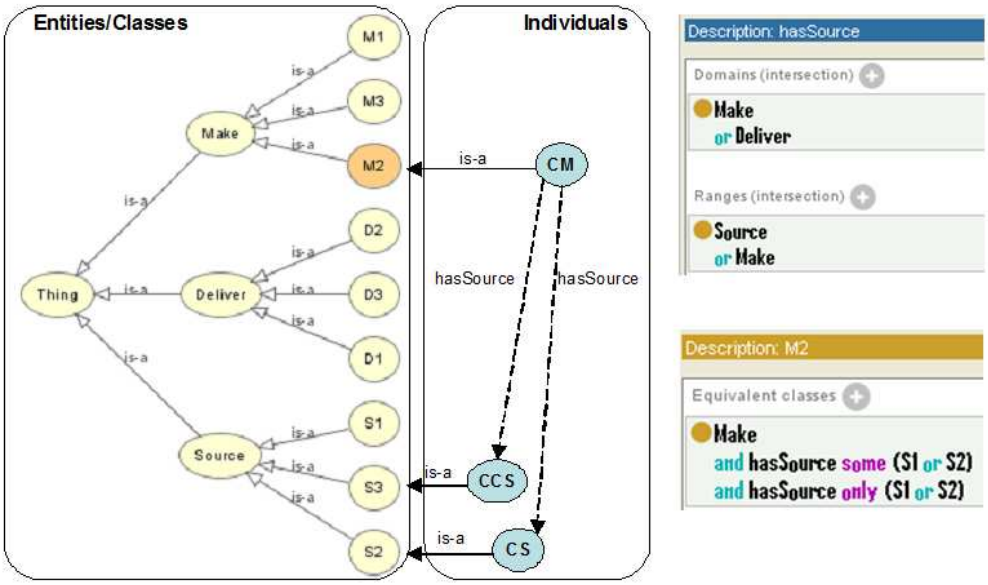
Representation of SCOR/OWL concept
216x129mm (96 x 96 DPI)



An example of SCOR/OWL graph
166x78mm (96 x 96 DPI)

```xml
<owl:ObjectProperty rdf:about="Is_The_Entry_For">
<rdfs:domain rdf:resource="#Production_Plans"/>
<rdfs:range rdf:resource="#ScheduleProductionActivities_(m_t_s)"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="Is_The_Entry_For">
<rdfs:domain rdf:resource="#Scheduled_Receipts"/>
<rdfs:range rdf:resource="#ScheduleProductionActivities_(m_t_s)"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="Has_An_Output">
<rdfs:domain rdf:resource="#ScheduleProductionActivities_(m_t_s)"/>
<rdfs:range rdf:resource="#Production_Schedule"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="Is_Measured_By">
<rdfs:domain rdf:resource="#ScheduleProductionActivities_(m_t_s)"/>
<rdfs:range rdf:resource="#Cost_to_Schedule_Production_Activities"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="Is_Measured_By">
<rdfs:domain rdf:resource="#ScheduleProductionActivities_(m_t_s)"/>
<rdfs:range rdf:resource="#Schedule_Achievement"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="Is_Needed_By">
<rdfs:domain rdf:resource="#Utilize_off_peak_shifts"/>
<rdfs:range rdf:resource="#ScheduleProductionActivities_(m_t_s)"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="Is_Needed_By">
<rdfs:domain rdf:resource="#Cross_Training_Certification"/>
<rdfs:range rdf:resource="#ScheduleProductionActivities_(m_t_s)"/>
</owl:ObjectProperty>
```

An example of SCOR/OWL code
151x162mm (96 x 96 DPI)



Example of incoherency detectable in SCOR/OWL
205x123mm (96 x 96 DPI)

An ontological approach for strategic alignment: a SCOR case study

O. Sakka*, P-A. Millet, V. Botta-Genoulaz

INSA-Lyon, LIESP, F-69621 Villeurbanne cedex, France

Postal address: INSA-Lyon, DISP-LIESP, bât Léonard de Vinci, 17, Av. Jean Capelle, 69621 Villeurbanne cedex, France
Phone: +(33) (0) 4.72.43.62.91
Fax: +(33) (0) 4.72.43.83.14
E-mails: omar.sakka@insa-lyon.fr, pierre-alain.millet@insa-lyon.fr, valerie.botta@insa-lyon.fr

(Received XX Month Year; final version received XX Month Year)

Firms cannot be competitive if their business and information technology strategies are not aligned. Yet achieving strategic alignment continues to be a major concern for business executives. A number of alignment models have been proposed in the literature. Enterprise Modelling (EM) can deliver models that are understandable by all participants and formalised enough to map the Enterprise Engineering and Reengineering activities directly onto the business process execution. However, models need terms (names, verbs...) to identify and describe the constructs modelled in the EM language used. To share business knowledge, a common modelling language is not sufficient. A common business language is required to share the understanding of any constructs used in the modelling language at a semantic level. The aim of this paper is to present the importance of knowledge formalization for strategic alignment. Our work is based on knowledge contained in a well-known reference model for supply chain: SCOR model. To analyze this knowledge, we transform this model into ontology. Finally, we will explore the respective advantages of the different representations of SCOR model (original text, using a business modelling language, ontology), and more generally, the contribution of ontologies as they are becoming a major issue in business modelling.

Keywords**:** Strategic alignment, Business Process Modelling, SCOR model, Supply Chain, OWL.

1. Introduction

Efficient Supply Chain Management (SCM) requires consistent exchange and sharing of information, which is often hindered by semantic clashes among heterogeneous applications. The literature suggests that firms cannot be competitive if their business and information technology strategies are not aligned. Yet achieving strategic alignment continues to be a major concern for business executives. To achieve this, a number of alignment models and standards have been proposed in the literature.

Furthermore, most enterprise application integration standards have adopted eXtensible Markup Language (XML) technologies to support the information integration of web-based SCM systems. XML standardizes the syntax of information exchange by defining mark-ups and structures of documents using tags. However, XML documents fail to capture the semantics (meaning) of data (Ray and Jones, 2003). To share business knowledge, a common business language is required for sharing, at a semantic level, the understanding of any constructs used in the modelling language. This business knowledge is now often formalized as a "reference model" which allows a "standard-based" business modelling for a common acceptance of the models. They are defined in modelling frameworks as "generic enterprise models" (Bernus and Nemes, 1996).

Deleted: the

In addition, ontologies are becoming a main issue of business modelling for the use of reference model in engineering projects. This allows building specific models instantiated from generic models in a specific business context. Benefits of using ontologies include reuse, sharing and portability of knowledge across platforms, as well as improving documentation, maintenance, and reliability (Uschold and Jasper, 1999).

In our work, we will attempt to present the importance of knowledge formalization for strategic alignment. We will base our work on a well-known reference model for supply chain: the Supply Chain Operations Reference[1] (SCOR) model from the Supply Chain Council. The ability to use such a reference model in an engineering project is a typical case study for semantic alignment. Our main contribution is a characterisation of issues for transforming a business reference model in a business ontology: we use the Web Ontology Language (OWL) proposed

[1] www.supply-chain.org/

by the semantic Web community[2] (W3C) so as to represent the SCOR model in an OWL ontology. Then, the OWL ontology can help highlight ambiguities between business partners, thus improving business alignment. Different versions of the SCOR model have been presented as texts. Therefore, representing the SCOR model in an OWL ontology requires formalising the current text version.

The remainder of this paper is organized as follows. The next section describes the literature related to strategic alignment. The SCOR model is introduced in Section 3. In Section 4, we will present the transformation methodology used to obtain our OWL ontology of the SCOR model. We will illustrate our approach by an application in Section 5. Section 6 emphasizes the contribution of this paper by putting it in context. Section 7 concludes.

2. Literature review

In this section, we will first expose strategic alignment concepts, specifically the strategic alignment model (SAM) (Henderson and Venkatramen, 1989). Then, we will focus on knowledge dimensions in SAM. Considering this semantic challenge, we will present ontologies as potential approaches for alignment. Finally, we will position this issue in the field of supply chain interoperability, based on the SCOR model.

The literature suggests that firms cannot be competitive if their business and Information Technology (IT) strategies are not aligned. The strategic alignment of IT exists when business organization's goals, activities, and processes are in harmony with the information systems that support them (McKeen and Smith, 2003). Achieving strategic alignment continues to be a major concern for business executives. Strategic alignment positively influences IT effectiveness (Ciborra, 1997),

[2] http://www.w3.org/TR/owl-features/

leading to greater business profitability (Luftman et al., 1999). Some studies (Labovitz and Rosansky, 1997; Corrall, 2000; Galliers and Newell, 2003) highlight the importance of strategic alignment. In contrast to some other areas of IT research, the literature provides little guidance on how to achieve alignment between business and IT strategies. Moreover, there is debate in the literature about what alignment actually is, why it is required, and how firms may organize their alignment process.

As a consequence of this debate, strategic alignment has many pseudonyms (Avison *et al.* 2004). We can quote "linkage" (Henderson and Venkatramen, 1989), "harmony" (Luftman, Papp et al., 1999), "bridge" (Ciborra, 1997), and "integration" (Weill and Broadbent, 1998). The common point is the requirement to relate business strategy and IT. In this sense, cooperation between the business and the IT department to maximize investment in technology is vital, and with this in mind, IT investments and business objectives have to be considered together (Avison et al., 2004).

The concept of strategic alignment is seen in various ways. Although Jarvenpaa and Ives (1994) argue that, an over-tight fit between IT and business strategy may reduce strategic flexibility, Smaczny (2001) asserts that IT is pervasive in business and should not be considered as separable from business strategy. Therefore, the need for alignment does not arise.

Having argued that alignment is usually viewed as beneficial, the following relates to how firms may become aligned. It is important for firms to choose the appropriate way and strategy to attain alignment. Bleistein *et al.* (2005, 2006a) present a requirement analysis approach for verification and validation of requirements in terms of alignment with and support for business strategy. This approach combines the use of business strategy analytical tools and requirement engineering techniques. Bleistein *et al*. (2006b) demonstrate how to scope the context of a strategic organizational IT requirement problem. They based their work on the strategic business modelling framework proposed by (Weill and Vitale, 2001) to validate the alignment of requirements between business strategy and business processes.

According to Luftman *et al.* (1999), understanding the alignment process leads to consider what may enable or inhibit it. These authors quote some enablers (executive support for IT, starting development in tandem…) and some inhibitors (no close relationship between the IT department and the business, the IT department neither knows its customers nor meets its commitments, resulting in little executive support for IT…).

For Papp (1999), alignment considers both the links between strategy and infrastructure, and a fundamental integration between business and IT.

More often, after the alignment achievement, environmental changes may reduce this alignment because of over-emphasis, complacency and inertia, engendering a need for revolutionary changes (Avison *et al.*, 2004). Their results demonstrate that some firms have low alignment or misalignment even during evolutionary periods. A number of models of strategic alignment have been proposed. Two approaches have attracted most attention from researchers, the MIT'90 (Morton, 1991) and the Strategic Alignment Model, SAM (Henderson and Venkatramen, 1989). According to Morton's MIT'90 framework (Morton, 1991), IT value is best captured when aligned with business strategy, management processes, organizational structure, and individuals and roles in the organization. Henderson and Venkatramen (1989) recognize, in their Strategic Alignment Model, the potential of IT to both support and shape business policy. They also elevate IT from the traditional role of a support mechanism to a strategic role of business enabler. In this sense and in comparison with MIT'90 model, SAM makes a distinction between the external perspective of IT (IT strategy) and the internal focus of IT (IT infrastructure and process). This distinction implies two levels of integration: strategic integration between IT and business strategy, which establishes the capability of IT at a strategic level, and operational integration, the link between IT infrastructure and process and organizational internal infrastructure and processes. SAM is based on the alignment of four domains: business strategy, IT strategy, organization infrastructure and processes, and IT infrastructure and processes. In order to support this alignment, (Gudas et al., 2006) extended SAM by adding a knowledge-management level represented by the two boxes in the middle of Figure 1, i.e., Business and IT knowledge management. These authors propose that the company relies on its central knowledge base, shared by the four original domains represented in the four corners. They think that, for strategic alignment, the alignment of knowledge should become a central focus.

We will focus on this assumption to present the importance of knowledge formalization for strategic alignment. We are interested in the semantic techniques based on ontologies because they lead to a better understanding of a field and to more effective and efficient handling of information in that field. Because knowledge may have different representations, we nowadays find several ontologies of an area for the same field of application. Furthermore, applications may need to use ontologies from various areas or from different views on one area. On the other hand, ontology builders may want to use existing ontologies as a basis for the creation of new ontologies, by either extending these existing ontologies or combining knowledge from different smaller ontologies. In each of these cases, it is important to know the relationships between the terms in the different ontologies. It is, then, necessary to have tools allowing one to make the link between the knowledge expressed in each ontology. Thus, ontology alignment makes it possible to reconcile the opinions of several experts (Bach and Dieng-Kuntz, 2004) from a semantic point of view.

Euzenat *et al*. (2007) define "ontology matching" as the process of finding the relations between ontologies, and ontology alignment as the result of the process in which these relations are declaratively expressed.

Klein (2001) and Ehrig (2006) define the term "ontology alignment" as, given two ontologies A and B, finding for an entity of A, a corresponding entity in B, which has the same intended meaning.

These different approaches to ontology-matching or -alignment are currently used to solve the "semantic challenge". Some tools are being developed and some specific problems can be solved but, generally speaking, knowledge-mapping is a complex and unsolved problem. Our aim is to build a specific case, using normalized knowledge on supply chains drawn from existing business models, and to study how knowledge-mapping can help strategic alignment.

In the context of inter-organizational systems, alignment has to deal with integration of product, process, and information flows within and across organizational boundaries. The global competitive business environment, complex and rapidly changing customer demands, as well as advances in IT forced companies to look for efficiency not only in their internal operations, but also in their coordinated operations with their suppliers, partners and customers. Therefore, an alignment challenge is required to use effectively information technologies in order to identify, communicate and continuously improve extended enterprise processes. In this aim, the Supply Chain Operations Reference (SCOR) model is one of the best-known approaches to a technology-neutral enabler for process management.

The SCOR model allows one to establish the indispensable links between operational performances, financial results and strategic objectives. The model simultaneously, allows one to evaluate the effectiveness of the supply chain (SC), set up best practices and align the different elements of the SC. Building an OWL-based ontology of SCOR presents the advantage of expressing the knowledge of the SCOR model as a set of interrelated concepts that can be analyzed by means of ontological principles. This representation as an ontology enables one to study completeness and consistency of the SCOR model without scrolling the hundreds of pages of its text description. This helps us to understand the relations between some concepts included in the SCOR model, such as processes, best practices and performance attributes, and to verify that there are no contradictions or missing links.

3. SCOR reference model

The SCOR model was a grassroots initiative in the SCM. Around 69 industry practitioners founded the Supply-Chain Council (SCC) in 1996 as a professional forum on the emerging concepts of integrated management in the extended enterprise. The SCOR model became the SCC's key knowledge contribution to the field at a time when the functional barriers were still a challenge for the operations. The membership of the Council has reached 700, mainly consisting of practitioners, along with technology and consulting services providers, government and academic organizations. The SCOR model is one of the well-known business reference models (Stadler and Kilger, 2000; Huang et al., 2005; Roder and Tibken, 2006). It provides a terminology and standardized processes enabling a general description of SCs and their translation into process maps.

Rather than a vertical or technology-specific approach, the aim of the SCC is to produce a high-level process model. Consequently, SCOR may be applied to any product and information flow in the SC at a high-level of modelling, independently of detailed organisational or technological choices.

Level 1 of SCOR is the strategic level using five process types: plan, source, make, deliver and return (see Figure 2). This is the highest and the most aggregated level. Level 2 (tactical) is the configuration level using process categories (Make-to-stock, Make-to-order, etc.) taking into account SC typologies. In agreement with the company's strategy, this intermediate level makes it possible to (re)configure the SC from thirty sub-processes. Level 2 is the core level for SC modelling, level 3 (operational) is the core of process modelling, using process elements linked by information and physical flows. Companies using the SCOR model for SC evaluation or action can specify the activities of the sub-processes, the best practices, the information flows, the functionalities of software and tools. Level 4, which is not included in the reference model, is the implementation level and has to be defined specifically for each SC.

The SCOR model is used to improve knowledge and usage of the SCM system. It is more of a management framework. The SCOR model also has a number of additional abilities. It can function as a performance model with four levels, which are in turn shaped like a pyramid. These four levels can literally guide a company through each step of performance and through the model to analyze its SC. The SCOR model helps companies to identify SC problems, create a SC road map, and align business functions (Bauhof, 2004).

The model helps companies understand how the five processes are used repeatedly, along the whole chain from the suppliers of suppliers to the customers of customers. Each process is a critical link to manage a product flow successfully along the implementation level. The pyramidal structure of processes, from level 1 to level 4 helps to insure the consistency of the SC management, and to align the different views implemented in the model. An integrated and real-time enterprise SCOR view of an enterprise enables managers to better align SC applications with their business processes and strategic objectives, as well as support a more effective implementation of SCM process improvement initiatives (Gulledge et al., 2001).

The original SCOR model in version 9.0 is a textual bookmarked 650 pages file. It contains:

- 27 main processes (level 2) detailed in 171 elements of process (level 3),
- 274 information elements defined as input or output of process elements (level 3),
- 489 best practices identified at process levels 2 and 3, with 350 texts describing features of the information system required for each best practice,
- 498 metrics used at levels 1, 2 and 3, classified in 5 performance categories (cost, assets, reliability, agility, responsiveness).

To take advantage of the SCOR model and its knowledge, it would be interesting to model the original versions of SCOR in other more usable forms than a text.

4. Different representations of SCOR

4.1. SCOR original model

The SCOR model is composed of the following constructs:

- Process and process elements: standard descriptions of the individual elements that make up the SC processes;

- Metrics: standard definitions of key performance measurements;
- Best practices: descriptions of best practices associated with each of the process elements;
- Features: identification of software functionality that enables best practices;
- Input/output: identification of information exchanged between processes.

These elements represent a SCOR meta-model (see Figure 3), which highlights the alignment between business view (metrics, processes and practices) and IT view (information and features) of a SC.

This meta-model is a specific example of more generic enterprise modelling meta-models as the ISO 19440 norm. It can be mapped to this norm and then be considered as an illustrative case study of alignment model (Millet, 2008). Some SCOR constructs play a key role in the interaction between 19440 viewpoints: practices as organisational and resource viewpoints, and features as functional and resource viewpoints. This reinforces the importance of studying the SCOR model as an alignment tool.

With all these business concepts, the SCOR model organizes domain-specific business knowledge, defined in a modelling language, and is easily mapped to the ARIS language. Using only five modelling constructs, the SCOR model represents an extensive formalization of this knowledge into a business language, which is well known and recognized by supply chain experts.

4.2. SCOR/ARIS

ARIS is a framework designed by (Scheer and Schneider, 2005), which aims to represent a system according to five complementary views: functional, data, product/service view, organisational and process views. ARIS uses a modelling language with various kinds of models. The model for business process is known as Event-driven Process Chains (EPC), which is the centre of the “ARIS House”. It connects all other views, and describes the dynamics of the business processes. The

"ARIS House" represents a concept for comprehensive computer-aided Business Process Management (BPM). Furthermore, ARIS House was developed to implement business models in information systems. As ARIS connects applications with functionalities, tasks and organizational units, it already represents an approach that is closer to the requirement of alignment. There are links between the different points of view at an operational level. The value chain representation supports a top-down approach from strategic to operational alignment.

The SCOR reference model can be expressed thanks to the ARIS formalism and included in a database as explained above. There is not only one way to create a SCOR/ARIS model, as this transformation requires a certain interpretation of the SCOR original models, and some choices to consider the SCOR knowledge required in a modelling language. In the following, we describe the structure of a SCOR/ARIS model provided by the Supply Chain Council.

The SCOR/ARIS model is organized in six main "ARIS groups", and one dedicated to the legend that defines the objects selected to model SCOR with ARIS as summarized in Figure 4. The first three groups are dedicated to the description of the strategic (top), tactical (configuration) and operational levels (process category). The fourth directory comprises the performance attributes classified in five performance categories (cost, assets, reliability, agility and responsiveness). The fifth directory includes all the inputs and outputs of the processes recorded in a SCOR data overview model. Finally, the sixth directory groups together all the best practices.

The links between the constructs of the SCOR meta-model, i.e., process, metric, best practice, input and output, are expressed in the SCOR/ARIS model through a specific kind of model, the *Function Allocation Diagram* shown in Figure 5. Those links described in the diagrams are all included in the database and allow one to question the model with requests such as: what are the processes that require such a best practice to reach high performance attributes?

To build the SCOR/ARIS model, several choices have been made:

- SCOR/ARIS is structured in a hierarchy of groups, which gives a classification of the different elements. This organisation is in itself an interpretation of the original textual SCOR model index.
- The features of the original SCOR are not part of the SCOR/ARIS model. Since the version 8.0 of SCOR, the features, that used to be linked with the processes, have been included as attributes of good practices. Hence, they are not formalised enough to be included in the SCOR/ARIS model.

Figure 6 gives an overview of the processes in levels 1 and 2, represented in the ARIS formalism through a *Value-Added Chain*. The little sign in the bottom right corner of each box indicates that the object (i.e., the process) is described in less detail using one or several models, and allows easy navigation throughout the database. Hence, we can open two models from the “Make-to-stock” process box: a *Value-Added Chain* and a *Function Allocation Diagram*.

In the SCOR/ARIS model, there is no direct link between best practices and metrics. Best practices are related to metrics through processes. We can imagine that the adoption of one or several best practices for a particular process will have an influence on one or several metrics related to that process; but this influence is not obvious and has not been clearly formalised in the original model of SCOR. Nevertheless, in the original paper version of SCOR, an effort is made to improve the description of some of the best practices and to formalise how best practices influence metrics. As an example, in the original SCOR model, the impact of the best practice “Lean Methodology” is analysed on the global performance attributes such as reliability, responsiveness, agility, cost and assets. The impact of this practice on costs is described as follows: “*Cost reduction is a typical impact of lean as non-value added processes are removed and efficiency increases. Response times are also* *reduced which results in faster cash flow cycles*" (Stephens, 2001). From this description, we can deduce that there is a link between the "Lean Methodology" best practice, and the "Cost to make", "Make cycle time" and "Order fulfilment cycle time" metrics. Those attributes will be improved if this recommended best practice is applied to the "Make-to-stock" process (see Figure 7). However, this knowledge, textually described in the original SCOR model, is not included in the SCOR/ARIS model, as it has been interpreted from some textual knowledge of the original SCOR model.

4.3. SCOR/OWL

OWL is an ontology language for the semantic web, developed by the W3C organization. According to (Horrocks et al., 2003), OWL was initially designed to represent information about categories of objects and how objects are interrelated — the sort of information that is often called an ontology. OWL can also represent information about the objects themselves — the sort of information that is often thought of as data. It has three increasingly expressive sublanguages — OWL Lite, OWL DL and OWL Full. OWL can be used when the information contained in documents needs to be processed by applications, as opposed to situations where the content only need to be presented to humans. OWL can be used to explicitly represent the meaning of terms in glossaries and the relationships between these terms; in other words, OWL can represent ontologies. Therefore, using OWL ontologies to share knowledge among business partners should help them to better interpret, understand, and benefit from that knowledge.

Business modelling and ontology are two existing approaches to formalize enterprise knowledge. The SCOR model, like many business reference models was first described textually, with some illustrative diagrams. It was, then, represented using a business modelling language. We have used the ARIS representation of the SCOR model, as it can be process by computers. In this section, we will first show how we can extract knowledge from SCOR/ARIS to create SCOR/OWL. Therefore, the result version will be a subset of knowledge of the SCOR/ARIS version. Finally, we will compare these two versions and see the differences in knowledge management allowed by OWL in comparison with ARIS.

The transformation of SCOR/ARIS into SCOR/OWL can be summarised in the following steps:

- Export SCOR/ARIS from ARIS into an ARIS XML file;
- Translate this ARIS XML file in OWL to obtain an OWL XML file;
- Import this OWL XML file in Protégé to obtain the SCOR/OWL model. (Protégé is a free, open-source platform with a set of tools to construct domain models and knowledge-based applications with ontologies[3]).

Our contribution lies in the translator (second step). To express SCOR/ARIS knowledge in an OWL XML file, we have to decide, for each ARIS XML tag, depending of an ARIS modelling object, how to represent it as an OWL XML tag. The final result is a transformation rule of each ARIS XML elements into OWL XML elements. Despite the apparent simplicity of this transformation, the following critical questions were asked:

- The Document Type Definition of an ARIS XML file defines a rich structure including graphical characteristics such as colours, positions, etc., and authorisation information including owners, rights, etc. The aim of our transformation is to capture the knowledge contained in the SCOR model. Should all this information be considered as semantic contents? That is obviously not the case for font or character set for example. The graphical position of an object in a model, above or below another one, may contain some assessment by the model designer on some implicit relations between these objects. In our work, we consider neither the graphical information of models, nor model management information as authorisations, creation/update dates, etc...
- Since ARIS is a business modelling language, it uses a meta-model containing several types of constructs (components, functions, process interface, module processes, organizational units, entities, etc.). The questions are how we can

Deleted: However,

Deleted: t

Deleted: the

Deleted: interfaces

[3] http://protege.stanford.edu/

present this diversity of types into an ontology, using only the notion of concept, and what class hierarchy we can build from the ARIS meta-model.

- ARIS organizes knowledge using models, which are partial "views" on the knowledge, and groups which are a hierarchical structure of knowledge. We have to decide whether to consider ARIS models as knowledge elements or not. Taking them into account means that we consider that the decision of a designer to create an ARIS model contains a semantic signification. The "point of view" makes sense by itself. If we do not take models into account, we consider that, irrespectively of the models used to present the knowledge, all of the semantic content is included in the objects and relations presented in the models.
- ARIS uses a hierarchical structure using the "association" between an object and some models. An object may be detailed in others through a model. However, it can also be a contextual model, which does not represent a hierarchical structure. In our case, these associations may be implemented as a hierarchy of concepts in OWL. This implementation is specific to the SCOR model, but is not necessarily valid in other cases.
- Once the transformation rules to create OWL concepts and relations are clear, we have to decide how we will choose the types of relations (transitive, functional, symmetrical) to represent them in OWL. Moreover, some restrictions can be defined on concepts in OWL. For example, two concepts may be mutually exclusive. Unfortunately, this notion is not represented in ARIS. Of course, "Plan process" is a concept which is very different from the metric "cycle time". On the other hand, the concepts "Make to Stock" and "Make to order" cannot be considered as exclusive. They may both be used in a middle-sized company where the production process is often heterogeneous and variable, and is thus, both a « Make to stock » and a « Make to order ».

To answer these questions, we propose a meta-model of both ARIS and OWL implementations to explain clearly the transformation rules. We, thus, ensure that there is some genericity and extensibility in the designed transformation rules, even if they are partially SCOR model dependent. The ARIS meta-model presented in Figure 8 contains "elements" which have many attributes.

As explained above, we will not take into account attributes regarding graphical format or model management. Figure 8 shows that elements are groups (used for the organization of objects in the ARIS application), models (used as groups of objects presented from a certain view), objects (which support the main semantic content), and object occurrences called ObjectOcc (which are the occurrence of an object in a model). Groups are composed of groups, objects and models. Models are composed of object occurrences. An object can be associated with a model (which is a particular description of the content or the context of this object). An object occurrence is linked to the reference object. For a better understanding of this meta-model, we add a rule contained in the ARIS application, and not in the XML export, but which is essential to the model's consistency. For each kind of model, only some objects are allowed. Then, we add an association class linking models and an object occurrence. In a conceptual meta-model, we would have to highlight the relations between objects, but in the ARIS XML file, these relations are implemented as attributes of objects.

The OWL meta-model described in Figure 9 is made of OWL classes and properties[4]. Classes may be of different kinds (union, intersection, complement and enumeration). Properties may be data properties or object properties.

Comparing the meta-models in Figures 8 and 9, some answers to our questions may be proposed:

- ARIS objects and groups are defined as concepts in OWL. Considering groups as a concept can be discussed. It can be considered as information regarding the management of formalized knowledge, and not the knowledge itself. In the case of the SCOR model, we consider groups as a concept in order to be sure to capture all the semantic knowledge regarding the grouping of objects, for example, when grouping all metrics by categories of costs, assets, reliability, etc. We consider that this decision is not SCOR dependant. Generally, model designers propose a hierarchical classification that helps to understand the structure of the knowledge embedded in the model.
- ARIS associations are defined as a hierarchy of classes in OWL. Considering the SCOR model, these associations are clearly used as a hierarchical description. Each level 2 process has, for example, a Function Allocation Diagram model, which describes the content of this process in terms of level 3 process elements. This hierarchy is clearly specific to the SCOR model. More generally, association can be used to link different points of view on the same concept, without any hierarchical structure between these views. For example, the same object can be associated with a "Function Allocation Diagram" model to describe a functional view, and with an "entity-relation" model to describe an informational view.

[4] http://www.w3.org/TR/owl-features/

- ARIS models will not be considered as concepts, but only as OWL objects. Therefore, models are not considered as an element of knowledge but only as a point of view on the knowledge. We consider this rule as a first step. If a specific view on some objects contains more knowledge than the objects themselves, this is an enrichment which cannot be decisive for the understanding. Semantic content of a model is necessarily an "add-on" to the semantic content of objects.
- All ARIS relationships between ARIS objects will be represented as OWL object properties. Other ARIS object attributes are transformed into OWL data properties.

4.4. SCOR/ARIS to SCOR/OWL transformation process

To transform a SCOR/ARIS XML file to a SCOR/OWL file, we used the eXtensible Stylesheet Language Transformations[5] (XSLT). XSLT is an official recommendation of the World Wide Web Consortium (W3C). It provides a flexible, powerful language for transforming XML documents into different documents types (HTML, XML PDF…). XSLT documents are well-formed XML documents that describe how another XML document should be transformed. For XSLT to work, it needs an XML document (the SCOR/ARIS document in our case) and template rules to make the transformation. In addition, parameters can be passed into XSLTs providing further instructions on how to do the transformation. Figure 10 describes the transformation process.

We have developed a XSLT-based tool to achieve the process transformation using the Orangevolt XSLT[6] plug-in on the Eclipse platform[7].

We have applied the transformation process twice. The first transformation is to obtain a reduced XML file that contains only information representing knowledge embedded in the model. For example, graphical data, creation/update date,… and even some modeling objects or attributes like "models" for reasons explained in §4.3.

[5] http://www.w3.org/TR/xslt
[6] http://eclipsexslt.sourceforge.net/
[7] http://www.eclipse.org/

Deleted: e

Formatted: Font: 10 pt

Formatted: Footnotes

Deleted: ¶

The example shown in Figure 11 describes an XSLT rule to select the second level of a hierarchy of groups.

The second step is to transform the new XML file SCOR/ARIS into OWL. Thus, we defined the XSLT rules to search for different types of relationships between objects of SCOR / ARIS, transform objects and groups to OWL concepts, and create the hierarchy of the SCOR / OWL file. Figure 12 shows an XSLT rule to manage the relationship type "Is Needed By" and figure 13 shows an example of creating tags of the OWL file SCOR / OWL.

5. Illustration and discussion

5.1. The three forms of SCOR model

We may distinguish three forms of representations of the SCOR model, as summarised in Table 1:

- Original SCOR (SCOR/text). The textual model of SCOR proposed by the Supply Chain Council is the reference for our work. We consider this model as the complete knowledge standardised by business practitioners. We investigate the advantages and drawbacks that other kinds of representations may have in comparison with this textual model.
- SCOR/ARIS. We call SCOR/ARIS the representation of SCOR using the ARIS language, a representation provided by the Supply Chain Council. This language uses a proprietary meta-model described in Figure 8. This representation provided by ARIS is not the only one of its kind. Some efforts have also been made to normalize knowledge embedded in the SCOR/text using only a spreadsheet to manage lists of concepts. Other representations using modelling tools exist. In all cases, the transformation is an interpretation of the knowledge. The capability of ARIS to express various points of view in modelling is a key-point to make this SCOR/ARIS as a reference for SCOR model computerizable representation. This capability may contribute to an alignment study between different points of view.
- SCOR/OWL. We call SCOR/OWL, our version of SCOR/ARIS transformed into an OWL representation. This transformation is discussed in Section 4.3. Consequently, the semantic included in the SCOR/OWL version is a subset of the semantic in SCOR/ARIS. Figure 14 shows the different concepts of SCOR/OWL presented in Protégé.

Deleted: computable

To compare the SCOR/ARIS model with its SCOR/OWL representation, we present an example of level 3 of the "make to stock" process: "production schedule activities". Relations between input/output, best practices, metrics and the "schedule production activities" process are presented as a graph of concepts in Figure 15 and as an OWL file generated by our translator in Figure 16.

5.2. The Supply Chain Example

Using these generated OWL concepts and properties, we consider the ability of the SCOR ontology for support reasoning on SCOR knowledge. For example, a reasoner can test whether or not one class is a subclass of another class. By performing such tests on the classes into an ontology, a reasoner can compute an "inferred" ontology class hierarchy. To use that, we need to complete the SCOR/ARIS with constraints, which are not formalized in SCOR/ARIS but described, even implicitly, in SCOR documentation. For example, the S3 process (Source Engineer-to-Order) must supply an M3 process (Engineer-to-Order) and cannot supply an M1 process (Make-to-Stock). The consequence is that, from a semantic point of view, some SCOR/ARIS relations are restrictions to be defined in OWL as properties. In the previous example, a restriction can be expressed as follows: "a to-order process has to supply a to-order and not a to-stock process" or "a to-order process has as its source a to-order or a to-stock process". However, because this type of constraint is not formalized in the ARIS meta-model, we have to add it manually in SCOR/OWL, or to define some specific transformation rules to create OWL restrictions, depending on the ARIS relations. This is a clear example that a generic transformation of an ARIS model into an OWL model is not sufficient. For each ARIS model an expert has to define the restrictions that can be added, at the meta-model level, and even sometimes at the detailed model level. The properties added in the SCOR/OWL are used to classify concepts of a custom SCOR model and to check inconsistency of this custom model. For example, we can check that an instance of a process is not a "Make To Order"

Deleted: For

process, because of a particular property which is not allowed in the SCOR/OWL for this "Make To Order" concept.

The Figures 17 illustrate a supply chain example, a small case using three processes created as individuals added to the SCOR ontology. The reasoner is able to detect inconsistencies in this case. SCOR/ARIS provides the tree of processes of levels 1, 2 and 3 in the left part of figure 17 (e.g., Deliver, Make, Make-to-Stock, etc.). The case is in the right part of figure 17 and includes one Make-to-order process (called CM for Case Make), which is linked to two Source processes by a hasSource relationship (inverse of supply): one Make-to-order Source (called CS for Case Source) and one Engineering-to-Order Source (called CCS for second Case Source). the SCOR/OWL was enriched with restrictions such as:

Deleted: In order to allow for reasoning, t

Deleted: his ontology

- a Make-to-order process (M2) "hasSource" only a Source Stoked Product (S1) or a Source Make-to-order process (S2)
- a Make-to-order process (M2) "supplies" only the Deliver Make-to-order (D2) or Deliver Engineering-to-order process (D3).

These restrictions were manually added in the ontology, using two inverse OWL properties: hasSource and supplies. The left side of figure 17 shows the input and output of hasSource properties and the lower picture in the right of figure 17 illustrates the rule that gives a restriction between M2 (thus, its individual CM) and S1 or S2 by a hasSource property.

5.3. Comparison between different representations

We now compare, side by side, each one of the considered representations of the SCOR model. These representations are formally different, but share the same knowledge, and can be used by both a modelling engineer, and a business practitioner in an alignment project. Therefore, we consider a comparison from a user's point of view:

- SCOR/text vs. XML-based SCOR: We may refer to SCOR/ARIS and SCOR/OWL together as XML-based SCOR. In short, this comparison emphasises the drawbacks and advantages of knowledge that computers can understand. The advantages stem from the automation of knowledge management with XML-based SCOR while the drawbacks come from their incomplete representation of knowledge. This is the consequence of the XML modelling process, which has to decide from a textual description to exclude some imprecise content of the SCOR/text. For example, formalising some SCOR/text description requires external tacit knowledge. Concerning the advantages, much knowledge is already managed informally by computers (e.g., e-mails), but cannot be used by them. Computers, in particular, may store and manipulate (copy/paste, search for strings, etc.) electronic versions of SCOR in some textual format, even formatted in a spreadsheet, but cannot manipulate the knowledge inside. On the other hand, XML-based SCOR allows computers to pretend they understand the representation of the model of a supply chain. For that purpose, XML proposes tags, which can be used by others computers tools to infer new knowledge. With these tags, computers tools can check the consistency of some models of supply chains represented with OWL, or the consistency of the SCOR model with exchange standards such as EDI or OAGIS. On the other hand, representing the SCOR model in XML also has a few drawbacks, which all seem to come from the poorer information represented. In fact, formalising everything requires making everything explicit. Computers cannot handle implicit knowledge, unlike humans. Making knowledge explicit is always a filter process, which reduces the knowledge spectrum. This process removes some knowledge from the text because it cannot represent the total meaning of the textual SCOR model. For example the properties added in the example (see §5.2) can be considered as an "implicit" knowledge in the textual SCOR model, and has been "filtered" during the modelling process of transforming SCOR/text into SCOR/ARIS. In addition, interpretation may be required to transform the original SCOR model into some XML-based SCOR representation. In fact, formalising the SCOR model requires the removal of ambiguities. These are made obvious among people in different parts of a supply chain, who have their own interpretation of the original reference model. In summary, XML-based SCOR allows computers to carry out much more reasoning than humans could do with the original model, but with poorer information.
- SCOR/ARIS vs. SCOR/OWL: As noted above, both representations of the SCOR model are consistent. More precisely, SCOR/OWL is a subset of SCOR/ARIS, but with the ability to add semantic knowledge using the benefits of ontologies. The main difference is that SCOR/OWL is described by means of a more generic meta-model and is used to build domain specific knowledge. In contrast, SCOR/ARIS is described by means of a specific meta-model dedicated to business models, and is (sometimes) used to model standard business knowledge. The OWL meta-model is domain-independent while the ARIS meta-model is domain-dependent. In addition, SCOR/ARIS is mainly used in an enterprise engineering project, to support decision-making, change management, and communication, while SCOR/OWL is used in enterprise knowledge management projects, to support knowledge mapping and alignment. The SCOR/ARIS models are "user-oriented" while SCOR/OWL models are mainly "expert-oriented". SCOR/ARIS helps to make

Deleted: spreadsheet

Deleted: are

Deleted: forces one

Deleted: to make

Deleted: since c

Deleted: requires

Deleted: , as the filter

Deleted: in

> decisions on organisations, processes, and roles, while SCOR/OWL helps a knowledge expert to decide about specific concepts, glossaries, and classifications of a specific case, using standard knowledge of SCOR/OWL. Even if these two representations are computerizable, we use SCOR/ARIS to simulate and evaluate metrics, and we use SCOR/OWL to define and check concepts classification, first step towards adding semantic rules to the knowledge. In fact, both SCOR/OWL and SCOR/ARIS can use XML-based tools (XSLT transformations, etc.), but only SCOR/OWL can use OWL specific tools (reasoner, Protégé plug-ins such as the inference engine, etc.).

Deleted: computable

Deleted: validate

Deleted:

Deleted: by reasoners

Using one or two representations depends on the user's needs. Enhancement of the SCOR model knowledge is always informal, using mainly text to identify and describe concepts and relations. However, a computerizable version is required for engineering projects. Knowledge engineering is best supported by SCOR/OWL while organizational engineering requires SCOR/ARIS. Business models and ontologies are complementary for enterprise engineering. Alignment is a semantic issue requiring ontologies, and an informational issue requiring business models.

Deleted: computable

Conclusion

Effective and efficient supply chain management needs to integrate heterogeneous applications in supply chains based on common representation and exchange of information semantics. Current related researches are not adequate to address the problem because they do not deal with knowledge interoperability (Ye et al., 2008). A key contribution of this work is to transform the Supply Chain Operations Reference (SCOR) model into an ontology. We use the Web Ontology Language (OWL) in order to benefit from advantages of ontologies, which can make easier collaboration in a supply chain by transmission of information, and which can be used to assist the alignment of business processes with standard e-commerce (Seng and Lin, 2007). Such a modelling in OWL requires formalising the SCOR model from its original text version.

The ontology of the SCOR model presented in this paper is the first step of a work on the use of ontologies in supply chains, and on the strategic alignment problem. We have described the way we have created our ontology and compared it with two other representations of the SCOR model. This work is a basis for semantic alignment studies with the following approach; alignment requires business knowledge elicitation, and this knowledge can be contained in business model, e.g. the SCOR model. To properly interpret this knowledge, we can take advantages of using ontologies mapping or merging approaches and for that, we need to transform this model into an ontology. This formalization should also help to study the consistency of the SCOR model with other existing models previously transformed into ontologies, or study the alignment between the SCOR model and exchange norms. Finally, some research focused on an operational point of view of Supply Chain are using an ontology of SCOR as a starting point for simulations, by relying on a model built by and for the industry.

Deleted: can
Deleted: e
Deleted: then
Deleted: have
Deleted: from
Deleted: , an
Deleted: is
Deleted: good
Deleted: SC
Deleted: famous

Acknowledgments: We thank Rhône-Alpes regional council (France) for its financial support for the MES (Manufacturing Execution System) project as part of the cluster EDIT and for the COPILOTES2 project regarding performance and collaboration in Supply Chain, as part of the cluster GOSPI. We thank also Mohamed Nabil Hammami, who developed the translator during its internship in the LIESP laboratory in 2009.

Deleted: parser

References

Avison, D., Jones, J., Powell, P. and Wilson, D., 2004. Using and validating the strategic alignment model. *Journal of Strategic Information Systems* 13(3), 223-246.

Bach, T. L. and Dieng-Kuntz, R., 2004. On Ontology Matching Problems (For Building a Corporate Semantic Web in a Multi-Communities Organization). *Software Agents and Internet Computing*, 236-243.

Bauhof, N., 2004. SCOR Model: Supply Chain operations Reference Model. *Beverage Industry*, 95(8), 78.

Bernus, P. and Nemes, L.,1996. A framework to define a generic enterprise reference architecture and methodology. *Computer Integrated Manufacturing Systems* 9(3), 179-191.

Bleistein, S. J., Cox, K., 2005. Strategic alignment in requirements analysis for organizational IT: an integrated approach. *Proceedings of the 2005 ACM symposium on Applied computing*. 1300-1307

Bleistein, S. J., Cox, K. and Verner, J., 2006b. Validating strategic alignment of organizational IT requirements using goal modeling and problem diagrams. *Journal of Systems and Software* 79(3), 362-378.

Bleistein, S. J., Cox, K., Verner, J. and Phalp, K. T., 2006a. B-SCP: A requirements analysis framework for validating strategic alignment of organizational IT based on strategy, context, and process. *Information and Software Technology* 48(9), 846-868.

Ciborra, Claudio U., 1997. De profundis? Deconstructing the concept of strategic alignment, *Scandinavian Journal of Information Systems*, 9 (1).

Corrall, S., 2000. S*trategic Management of Information Services: A Planning Handbook*. London: Aslib/IMI.

Ehrig, M., 2006. *Ontology alignment: bridging the semantic gap*. Germany: Springer.

Euzenat, J., Mocan, A. and Scharffe, F., 2007. Ontology Alignments: An Ontology Management Perspective. *In*: Hepp, M.; Leenheer, P. de; Moor, A. de; Sure, Y.,ed. *Ontology Management, Semantic Web, Semantic Web Services, and Business Applications.*US: *Springer*, 177-206.

Galliers, R. and Newell, S., 2003. Strategy as data+Sense Making In: Cummings. *In*: S. Cummings and D.C. Wilson, ed. *Images of Strategy*. Oxford : Blackwell, 164–196

Gudas, Saulius, Brundzaite and Rasa, 2006. *Knowledge-based enterprise modelling framework*. Berlin, ALLEMAGNE, Springer.

Gulledge, T., Cavusoglu, T. and Kessler, T., 2001. *Aligning the Supply Chain Operations Reference (SCOR) Model with Enterprise Applications: Real-time Value Chain Intelligence* [online]. Policy Analysis Center – Enterprise Engineering and Policy Program. Available from : http://www.scs.com.au/data/thoughtlib/SCOR&theenterprise.pdf

Henderson, J. C. and Venkatramen, N., 1989. Strategic alignment: leveraging information technology for transforming organizations, *IBM Systems Journal*, 38 (2-3), 472-484.

Horrocks, I., Patel-Schneider, P. F. and van Harmelen, F., 2003. From SHIQ and RDF to OWL: the making of a web ontology language. *Web Semantics: Sci., Serv. Agents World Wide Web* 1: 7-26.

Huang, S. H., Sheoran, S. K. and Keskar, H., 2005. Computer-assisted supply chain configuration based on supply chain operations reference (SCOR) model. *Computers & Industrial Engineering*, 48(2), 377-394.

Jarvenpaa, S. L. and Ives, B., 1994. The global network organization of the future: Information management opportunities and challenges. *Journal of Management Information Systems*, 10 (4), 25-57.

Klein, M., 2001. Combining and relating ontologies: an analysis of problems and solutions. *Ontologies and information sharing*, 47: 53-62.

Labovitz, G. and Rosansky, V., 1997. *The Power of Alignment* [online]. Available from: http://therealnavy.com/Documents/7.%20CNO%20Book%203%20Power%20of%20Alignment.pdf.

Luftman, J., Papp, R. and Brier, T., 1999. Business and IT in Harmony: Enablers and Inhibitors to Alignment. *Communications of Association of Information System* 1(3), 1-33.

McKeen, J. D. and Smith, H., 2003. *Making IT Happen: Critical Issues in IT Management*. Wiley.

Millet, P.A., 2008. Une étude de l'intégration organisationnelle et informationnelle : Application aux systèmes d'informations de type ERP. Thesis (PhD) INSA de Lyon.

Millet, P.A. and Botta-Genoulaz, V., 2006. Un référentiel pour l'alignement des systèmes d'information aux processus logistiques. *MOSIM'06*. Rabat (Maroc).

Morton, S., 1991. *The corporation of the 1990s: Information Technology and Organizational Transformation*. USA: *Oxford University Press*.

Papp, R., 1999. Business-IT Alignment: Productivity Paradox Payoffs ?. *Industrial Management & Data Systems*, 99 (8), 367-373.

Ray, S. R. and Jones, A. T., 2003. Manufacturing interoperability. *In Proceedings of the 10th ISPE International Conference on Concurrent Engineering: Research and Application,* 535-540.

Roder, A. and Tibken, B., 2006. A methodology for modeling inter-company supply chains and for evaluating a method of integrated product and process documentation. *European Journal of Operational Research*, 169(3): 1010-1029.

SCC. 2009. *Supply-Chain Council, SCOR Supply Chain Operations Reference Model version 9.0* [online]. Available from: http://www.supply-chain.org/."

Scheer, A. W. and Schneider, K., 2005. ARIS-Architecture of Integrated Information Systems. *In*: P. Bernus, K. Mertins and G. Schmidt, ed. Berlin Heidelberg: *Handbook on Architectures of Information Systems*.Springer, 605-623. .

Seng, J. and Lin, W., 2007. An ontology-assisted analysis in aligning business process with e-commerce standards. *Industrial Management & Data Systems* 107(3): 415-437.

Smaczny, T., 2001. IS an Alignment between Business and IT the Appropriate Paradigm toManage IT in Today's Organisation? *Management Decision* 39 (10), 797-802.

Stadler, H. and Kilger, C., 2000. Supply chain management and advanced planning; concepts, models, software and case studies. *Berlin*: *Springer-*,.

Stephens, S., 2001. Supply Chain Operations Reference Model Version 5.0: A New Tool to Improve Supply Chain Efficiency and Achieve Best Practice. *Information Systems Frontiers* 3(4), 471-476.

Uschold, M. and Jasper, R., 1999. A Framework for Understanding and Classifying Ontology Applications. *workshop on Ontologies and Problem-Solving Methods (KRR5) Stockholm, Sweden*.

Weill, P. and Broadbent, M., 1998. *Leveraging the New Infrastructure*. USA: Harvard Business School Press.

Weill, P. and Vitale, M., 2001. *Place to Space: Moving to eBusiness Models*. Boston : Harvard Business School Publishing Corporation,.

Ye, Y., Yang, D., Jiang, Z. andTong, L., 2008. An ontology-based architecture for implementing semantic integration of supply chain management. *International Journal of Computer Integrated Manufacturing* 21(1), 1-18.

Table 1. The three representations of SCOR model

Name of the representation	**Form**	**Advantages**	**Source**
Original SCOR	Text	Complete	SCC
SCOR/ARIS	XML	Computerizable	SCC
SCOR/OWL	OWL	Computerizable	Us

Formatted Table

Deleted: Computable

Deleted: Computable

Table 1. The three representations of SCOR model

Name of the representation	Form	Advantages	Source
Original SCOR	Text	Complete	SCC
SCOR/ARIS	XML	Computerizable	SCC
SCOR/OWL	OWL	Computerizable	Us

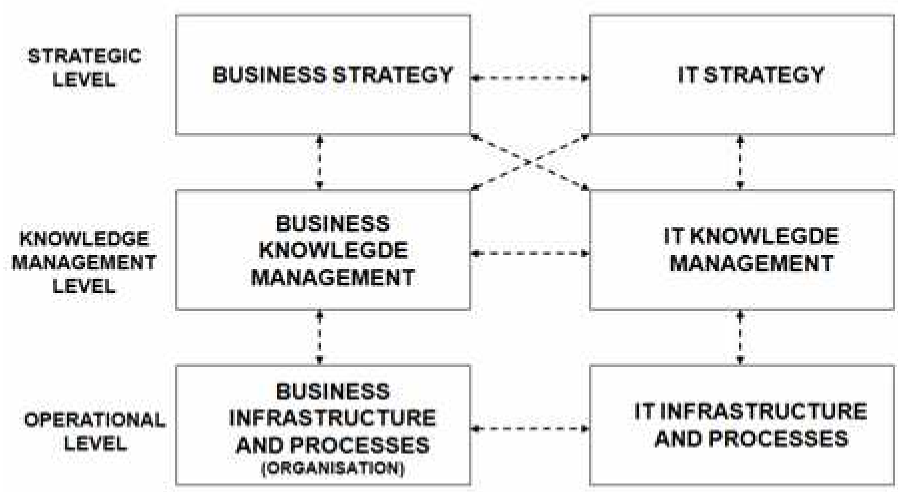


Figure 1. Strategic Alignment Model complemented with knowledge management components (Gudas, Saulius et al., 2006)

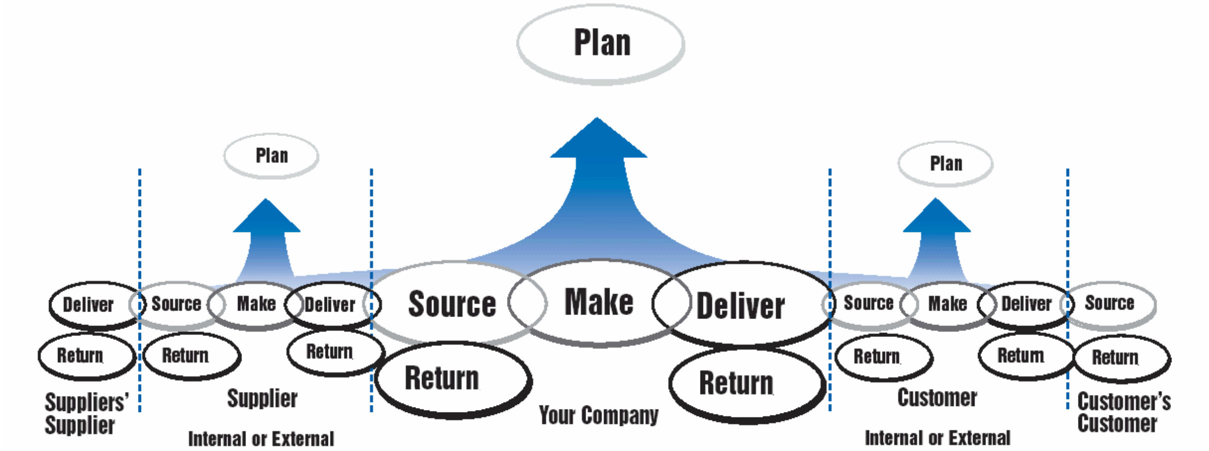


Figure 2. SCOR Model - Supply Chain Council (SCC, 2009)

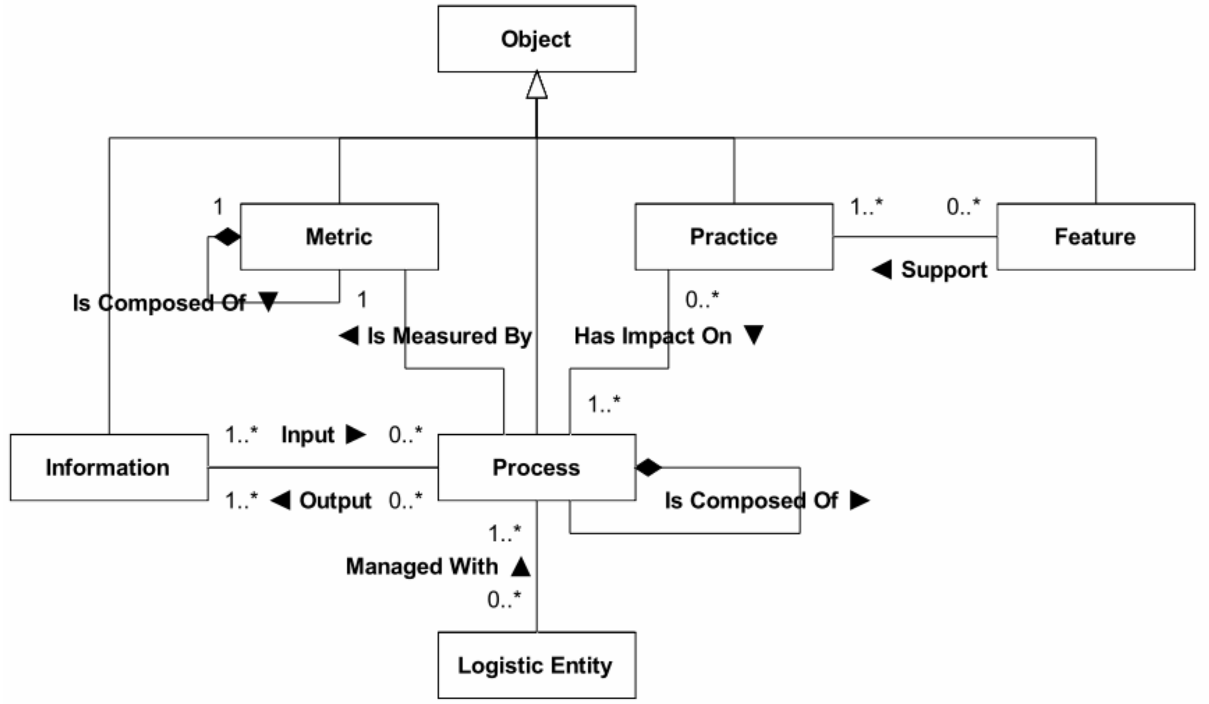


Figure 3. SCOR meta-model (Millet and Botta-Genoulaz, 2006)

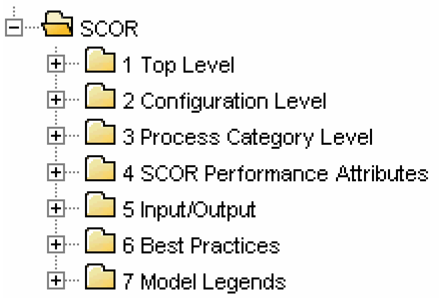


Figure 4. Tree of the SCOR/ARIS model

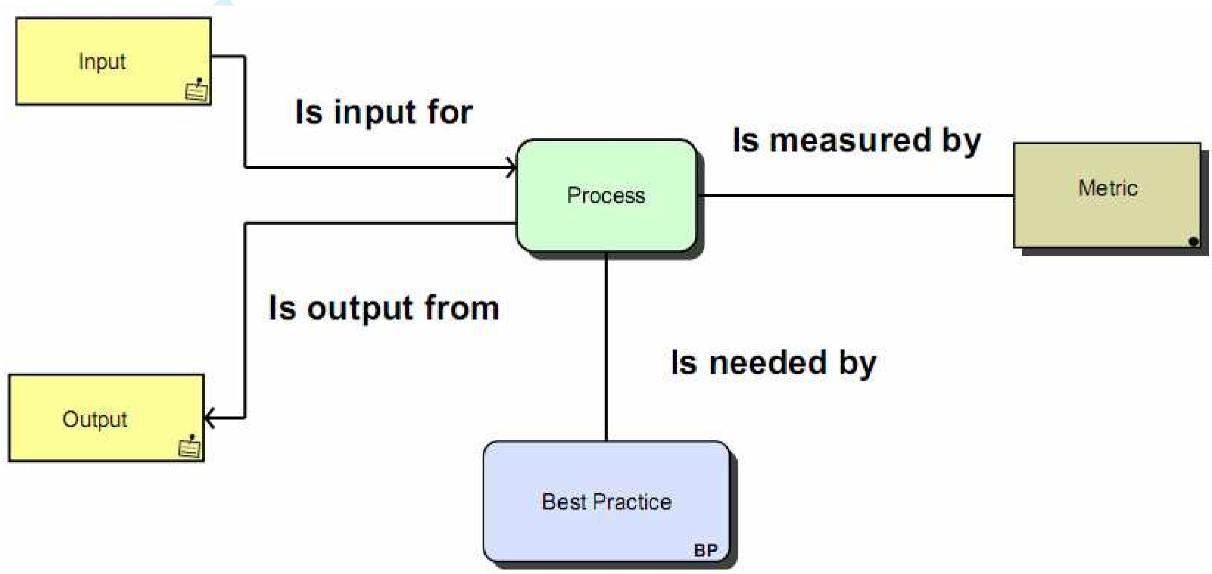


Figure 5. ARIS concepts supporting SCOR/ARIS

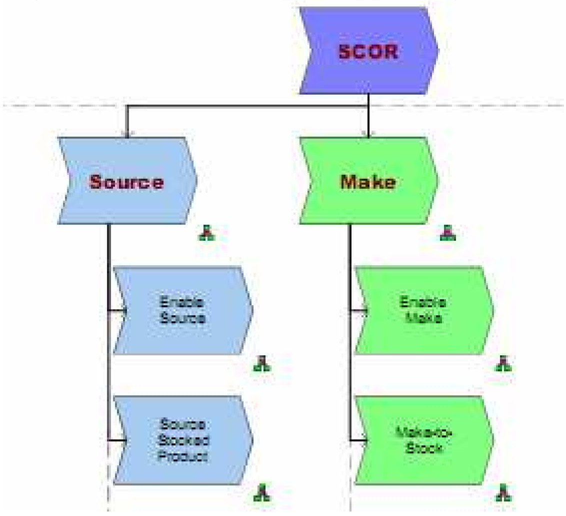


Figure 6. SCOR level 1 and 2 process overview

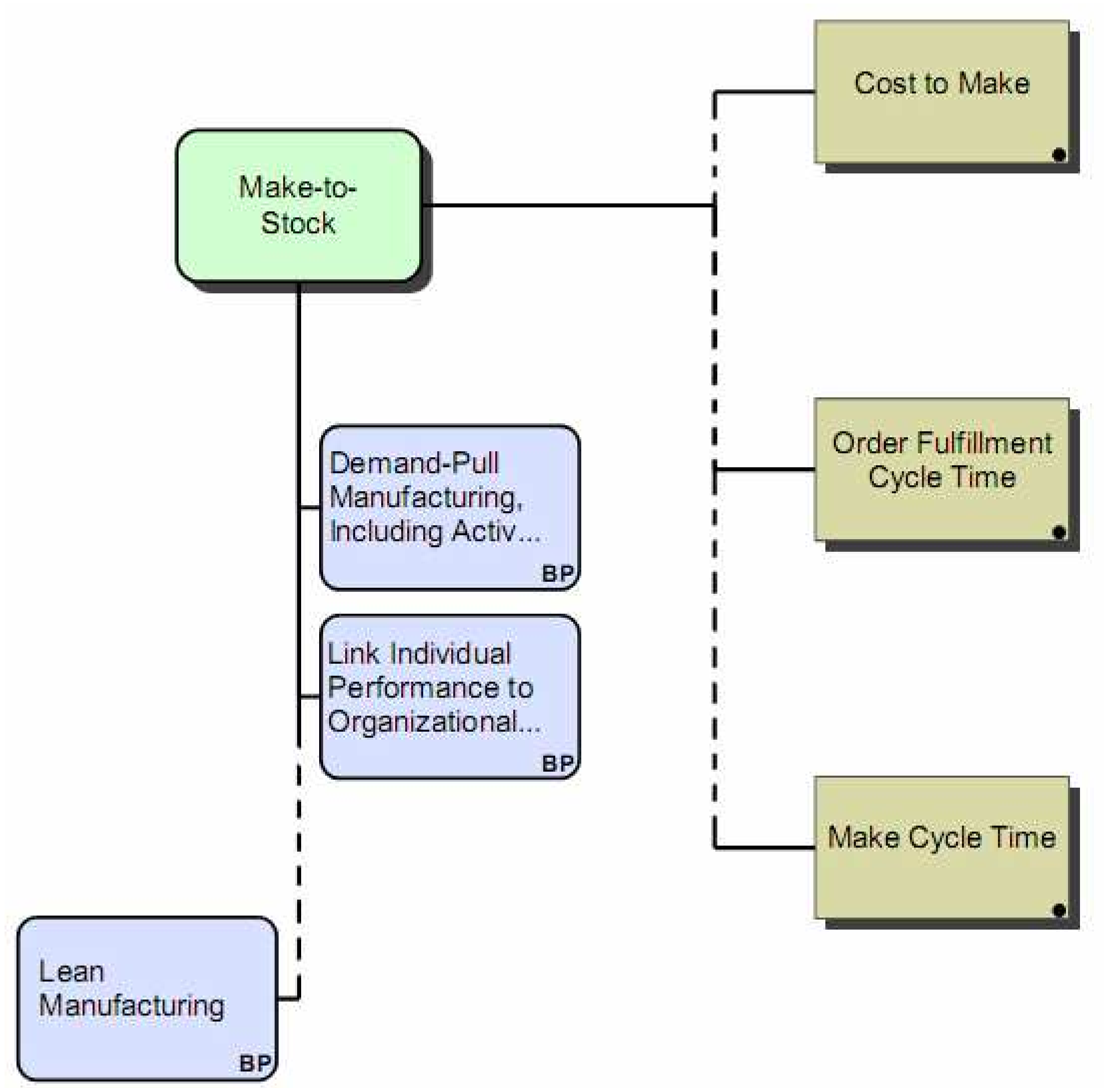


Figure 7. Function Allocation Diagram of the Make-to-stock process

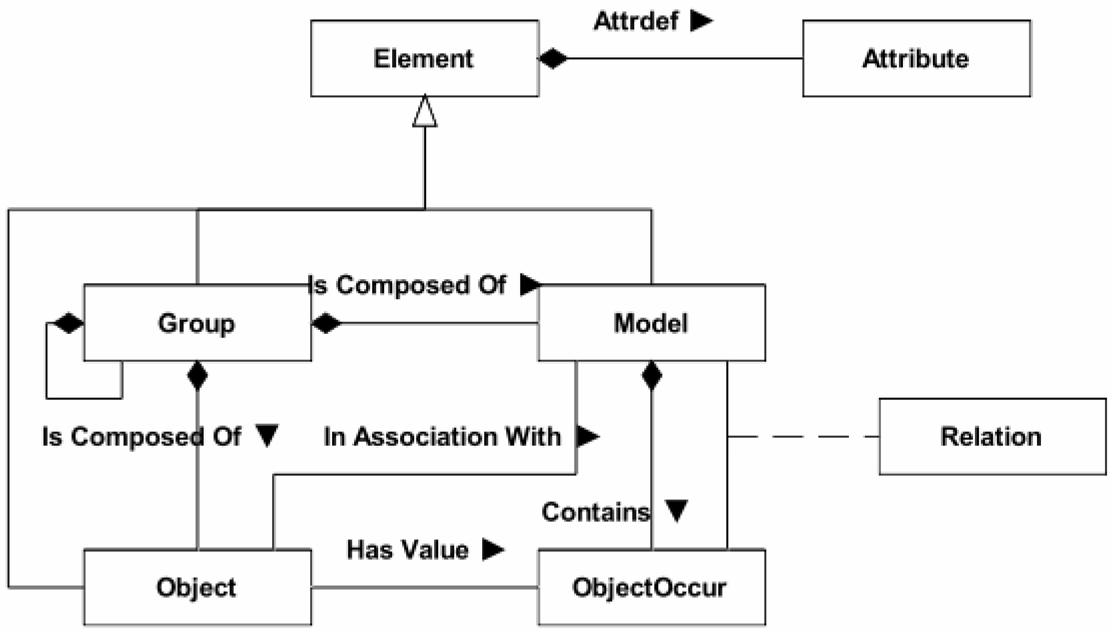


Figure 8. ARIS meta-model export

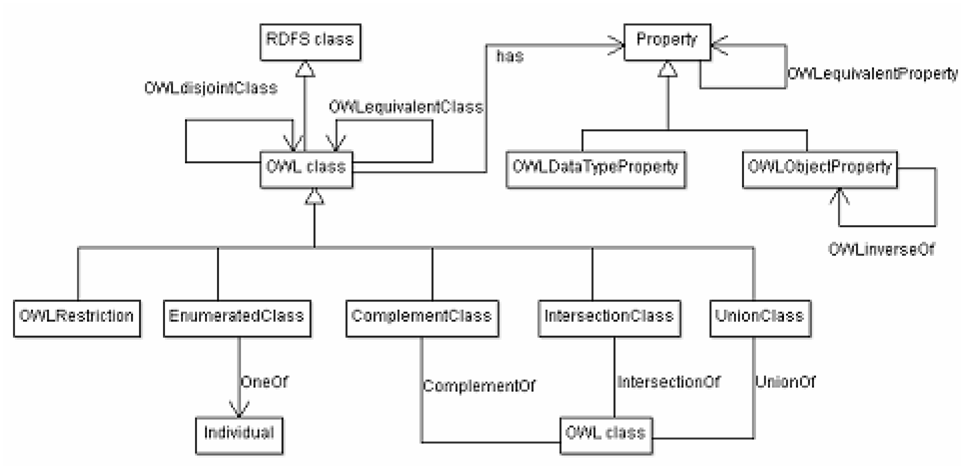


Figure 9. OWL meta-model

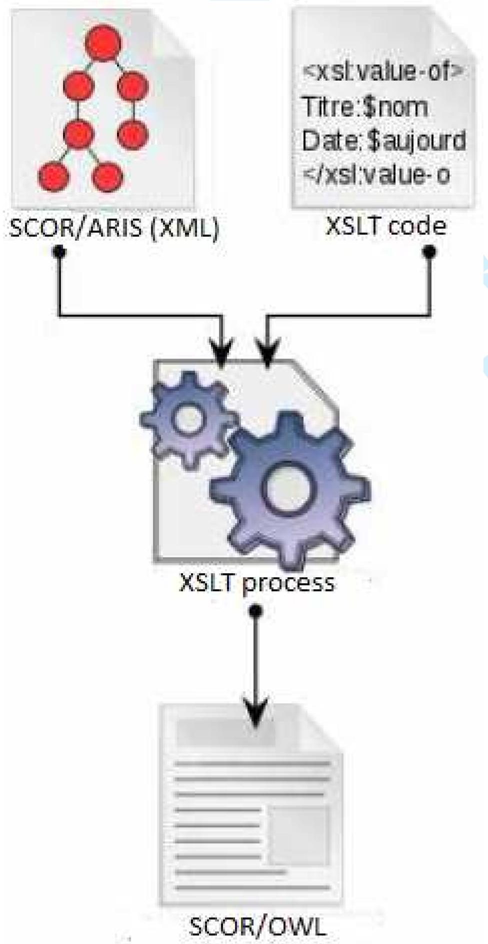


Figure 10. Transformation process

```
<xsl:for-each select="Group/Group">
<xsl:element name="Group1">
<xsl:text>&#10;</xsl:text>
 <xsl:call-template name="concept-group"/>
<xsl:text>&#10;</xsl:text>
<xsl:element name="Group2">
<xsl:value-of select="../AttrDef/AttrValue"/>
</xsl:element>
</xsl:element>
```

Figure 11. An XSLT rule

```
<xsl:when test="@CxnDef.Type='CT_IS_NEEDED_BY'">
<xsl:attribute name="rdf:about">
<xsl:value-of select="concat('Is_Needed_By','_',
(translate(../AttrDef/AttrValue,':;,?./§%µ-£
"<&>#|°}{@~-','_________________________')))" />
</xsl:attribute>
</xsl:when>
```

•

Figure 12.XSLT rule for “Is Needed by” relationship

```
<xsl:for-each select="Group/ObjDef">
- <xsl:for-each select="CxnDef">
- <xsl:element name="owl:ObjectProperty">
  <xsl:call-template name="recherche2" />
- <xsl:element name="rdfs:domain">
- <xsl:attribute name="rdf:resource">
  <xsl:value-of select="concat('#' , (translate(../AttrDef/AttrValue,':;,?./§%µ-£
"<&>#|°}{@~-','_________________________')))" />
  </xsl:attribute>
  </xsl:element>
```

Figure 13. XSLT rule for creating owl ObjectProperty

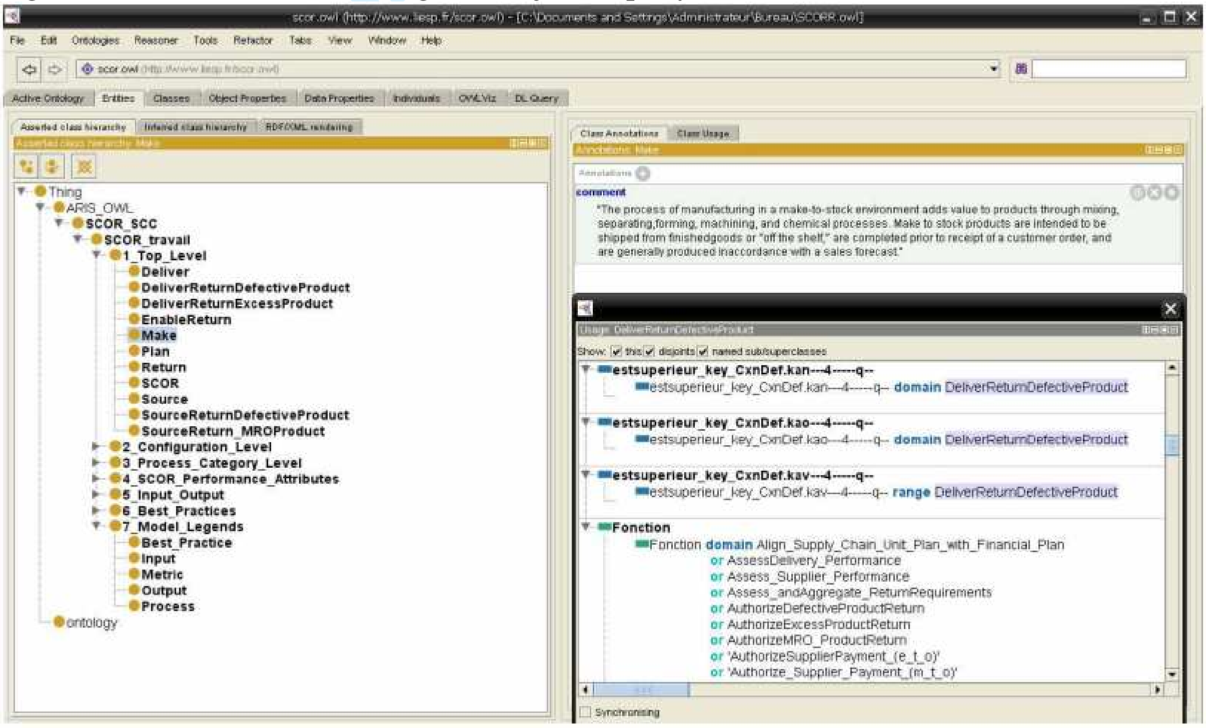


Figure 14. Representation of SCOR/OWL concept

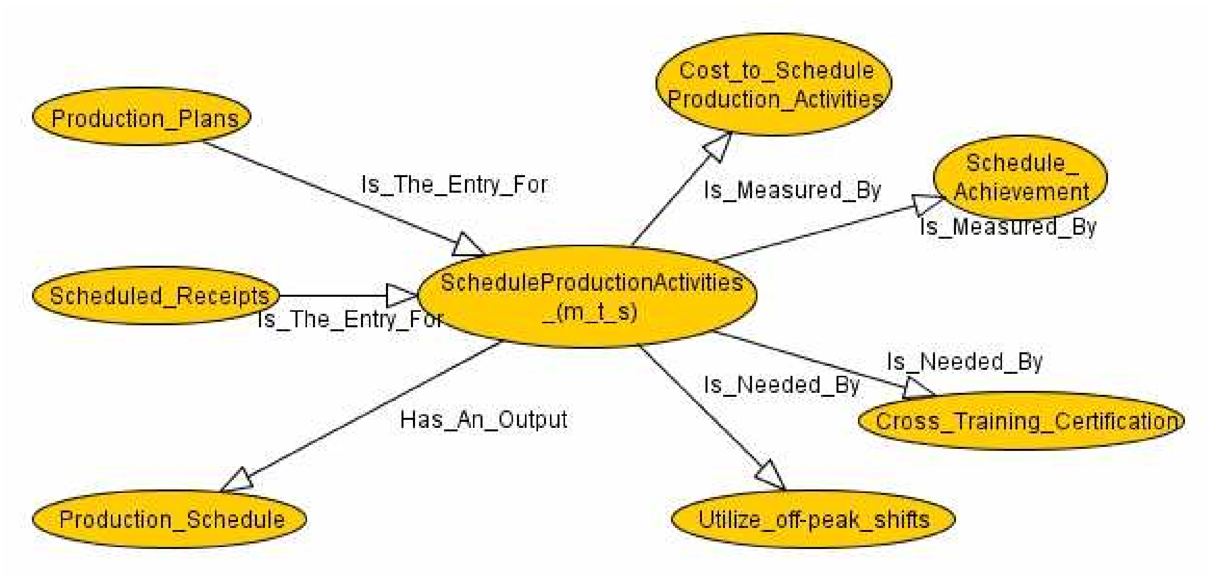


Figure 15. An example of SCOR/OWL graph

```
<owl:ObjectProperty rdf:about="Is_The_Entry_For">
<rdfs:domain rdf:resource="#Production_Plans"/>
<rdfs:range rdf:resource="#ScheduleProductionActivities_(m_t_s)"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="Is_The_Entry_For">
<rdfs:domain rdf:resource="#Scheduled_Receipts"/>
<rdfs:range rdf:resource="#ScheduleProductionActivities_(m_t_s)"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="Has_An_Output">
<rdfs:domain rdf:resource="#ScheduleProductionActivities_(m_t_s)"/>
<rdfs:range rdf:resource="#Production_Schedule"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="Is_Measured_By">
<rdfs:domain rdf:resource="#ScheduleProductionActivities_(m_t_s)"/>
<rdfs:range rdf:resource="#Cost_to_Schedule_Production_Activities"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="Is_Measured_By">
<rdfs:domain rdf:resource="#ScheduleProductionActivities_(m_t_s)"/>
<rdfs:range rdf:resource="#Schedule_Achievement"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="Is_Needed_By">
<rdfs:domain rdf:resource="#Utilize_off_peak_shifts"/>
<rdfs:range rdf:resource="#ScheduleProductionActivities_(m_t_s)"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="Is_Needed_By">
<rdfs:domain rdf:resource="#Cross_Training_Certification"/>
<rdfs:range rdf:resource="#ScheduleProductionActivities_(m_t_s)"/>
</owl:ObjectProperty>
```

Figure 16. An example of SCOR/OWL code

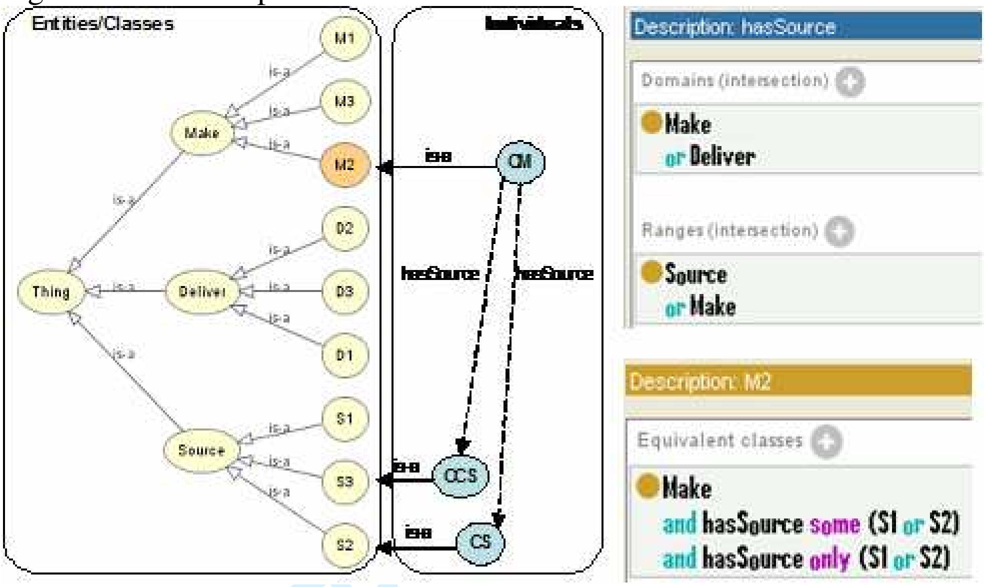


Figure 17. Example of incoherency detectable in SCOR/OWL

Dear professor,

We would like to thank you for your review and comments about our paper entitled “An ontological approach for strategic alignment: a SCOR case study”. Your comments help to tighten and enhance paper quality. We tried to apply them in our paper as possible.

In the following, we will answer some of your comment:

1/ XML and reasoning

Your remarks:

Please explain CLEARLY what you mean by this. As this reviewer remarked before, XML does not allow neither reasoning nor inference

XML DOES NOT allow inference!!!! Please, look for the help of a knowledge engineer.

Our answer:

Of course, XML noes not allow any inference by itself, but is a prerequisite to use reasoning tools...

In fact, it's not really XML, but a "processable by computers" formalization of knowledge. In our context, that means XML.

2/ SCOR/Owl and business expert...

Your remarks

What must the knowledge expert decide about ? If the OWL model is built, all the decisions about concepts, glossaries and classifications are already made ! !

Our answer

Even if a standard knowledge exists and is used by a practitioner or a business expert, this standard knowledge is discussed in the specific context of a particular enterprise.

That means that some concepts can be updated, specialized and become "customized" regarding the standard.

The relation between a standard knowledge and a particular case is not only based on "instantiation" of concepts in individuals, but also required to "adapt" the knowledge to the specific context.

3/ industrial application

Our Answer:

Our approach is currently being applied on another R&D project, where the MES (Manufacturing Execution System) business standard (ISA-95 / ISO 62264) is transforming into an OWL ontology (but this application was not the scope of this contribution).