On the Data Interoperability Issues in SCOR-Based Supply Chains

Milan Zdravković and Miroslav Trajanović

Laboratory for Intelligent Production Systems,
Faculty of Mechanical Engineering in Niš, University of Niš, Serbia
{milan.zdravkovic,traja}@masfak.ni.ac.rs

Abstract. Supply Chain Operations Reference (SCOR) is a reference model which can be used to design and implement inter-organizational processes of a supply chain. Its implementation assumes a high level of integration between the supply chain partners which reduces their flexibility. The problem of integration requirements may be addressed by enabling the supply chain partners to use their enterprise information systems (instead of specialized software tools) in the implementation and facilitation of SCOR processes. The performance of these processes can be significantly improved if the enterprise information systems of the supply chain actors are interoperable. In this paper, we are using semantic SCOR models to highlight data interoperability requirements for cross-enterprise SCOR processes and to make this data explicit, by relating it to the corresponding domain ontology concepts.

Keywords: Systems Interoperability, Ontology, Supply Chain, SCOR.

1 Introduction

Supply Chain Operations Reference (SCOR) [1] is a standard approach for analysis, design and implementation of five core processes in supply chains: plan, source, make, deliver and return. SCOR defines a framework which considers business processes, metrics, best practices and technologies with the objective to improve collaboration between partners. The SCOR model is implemented from the perspective of the single enterprise and it considers all interactions two levels ahead from the enterprise, towards its supply and customer directions. So, it assumes a significant level of cross-enterprise collaboration. This collaboration can be enabled by the specialized software tools or Enterprise Information Systems (EIS).

Implementation of SCOR reference model can be facilitated by the specialized software systems, such as ARIS EasySCOR [2] or e-SCOR [3]. However, the use of all of these systems implies a significant level of technical commitments of the enterprise and thus, it has a negative effect on their flexibility. Systems integration assumes fixed agreements on the message formats, interfaces and other types of commitments which implementation is costly and time consuming. In contrast to system integration, which basically deals with formats, protocols and processes of information exchange,

P. Herrero et al. (Eds.): OTM 2012 Workshops, LNCS 7567, pp. 154–161, 2012.

the objective of interoperability is to have two systems exchanging information with the consideration that they are not aware of each other's internal workings [4]. The main conditions for achievement of systems interoperability are: 1) to maximize the amount of semantics which can be utilized and 2) to make it increasingly explicit [5]. Then, this semantics can facilitate the interoperability at the different levels, such as data, processes and systems [6].

In this paper, we attempt to show how the above arguments can be used to resolve the data interoperability issues for the implementation of cross-enterprise SCOR processes. For this purpose, we are using implicit OWL representation of the SCOR model – SCOR-KOS OWL and its semantic enrichment – SCOR-Full [7], a micro theory which identifies and classifies common enterprise concepts in the context of supply chain operations. While they are explained in detail in the cited work, these representations are shortly described in Section 2 of this paper. In the Section 3, a main contribution of this paper is presented. The above mentioned formal models are used to infer about the data interoperability requirements in SCOR inter-organizational processes and to make this data explicit. The resulting formal representation can be used to facilitate the interoperability of two heterogeneous systems in context of the SCOR processes' requirements.

2 Formal Model of Supply Chain Operations

Although reference models reflect a common consensus of the industrial community about the specific domain, in most of the cases, they are developed by using a free form natural language. In such way, they are easily communicated throughout the community. However, the implicit definitions of the reference models entities make them difficult to exchange among EISs.

In our previous work [7], we made an attempt to address the problem of the required balance between implicit and explicit knowledge about the SCOR reference model. As a result, two logically related OWL formalizations of SCOR have been developed. First, implicit SCOR-KOS (Knowledge Organization System) OWL model is developed (see Figure 1). It directly translates the natural language form of SCOR entities to OWL language.

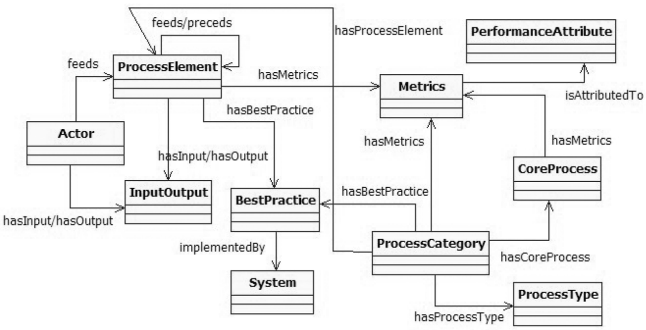

Fig. 1. Partial representation of SCOR-KOS OWL model

Second, a semantic analysis of SCOR entities is performed and SCOR-Full ontology is developed. SCOR-Full is a micro theory which formalizes knowledge about supply chain operations, by identifying and aggregating common enterprise notions. All concepts are classified into the generalizations, such as: Course, Setting, Quality, Function and Resource.

The implicit and explicit concepts of two models are then inter-related by using SWRL rules. Thus, it became possible to combine SCOR tools with other domain ontologies to make a formal reasoning about the process configuration [7], supply chain performance or database schemas [8] in context of SCOR reference model.

3 Data Interoperability Issues in SCOR Cross-Organizational Processes

SCOR reference model describes the key processes of supply chain operations (plan, source, make, deliver, return, enable) and their categories, according to the choice of the manufacturing strategy for a product (make-to-stock, make-to-order or engineer-to-order). These categories are then configured into cross-organizational processes.

For example, in the simple scenario, the relevant processes of the customer are plan source, source and return of the product or part. In this scenario, supplier plans manufacturing and delivery and subsequently make and deliver this product. Process categories can be decomposed into ordered set of process elements, each of which can exchange information with other process elements, within the same process category or externally.

SCOR-KOS OWL enables the inference of the relationships between individual process elements, namely, the flows of the tangible and intangible assets between activities of the processes. Hence, a direct reasoning about the data interoperability issues in SCOR cross-organizational processes can be carried out.

Figure 2 illustrates the exchange of the SCOR assets in the case of engineered-to-order manufacturing, between the customer and supplier (dashed lines). According to SCOR, the above mentioned manufacturing strategy assumes the exchange between “P2. Plan Source”, “S3. Source Engineered-to-Order Product”, “M3. Make Engineering-to-Order product” and “D3. Deliver Engineered-to-Order product” process categories. This process also involves following process categories: “EP. Enable Plan”, “ES. Enable Source”, “EM. Enable Make”, “ED. Enable Deliver” and “P3. Plan Make”. Only the latter process category from the last group of categories is illustrated on Figure 2, because of the visual representation complexity.

In the simple supply chain scenario, each of these process categories is assigned to a customer or a supplier.

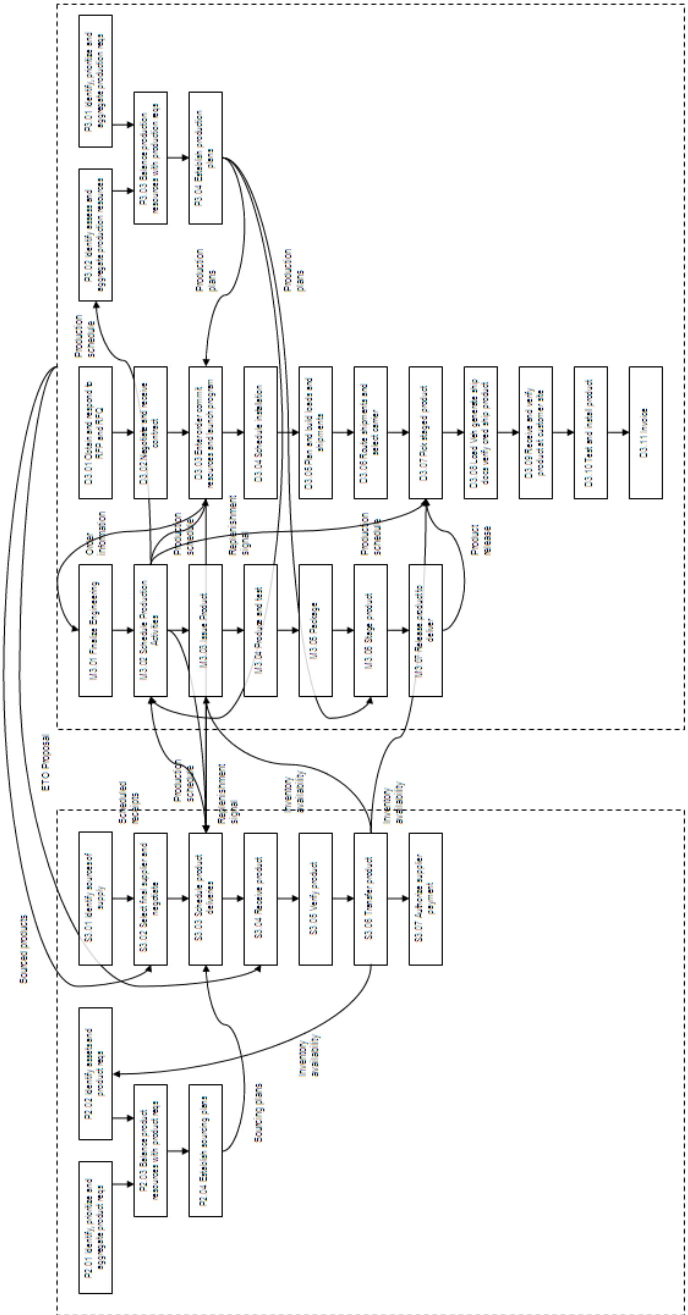

Fig. 2. Assets flows between process elements for engineered-to-order production type

If we assume that both partners are using the EISs, these systems can be considered as interoperable (in context of exchange information between SCOR processes), if they are capable to transmit and understand the information which is exchanged between following process categories:

- `S3_Source_Engineer-to-Order_Product` of the customer and `M3_Engineer-to-Order` of the supplier
- `S3_Source_Engineer-to-Order_Product` of the customer and `D3_Deliver_Engineered-to-Order_Product` of the supplier,

and in opposite direction:

- `M3_Engineer-to-Order` and `S3_Source_Engineer-to-Order_Product`
- `D3_Deliver_Engineered-to-Order_Product` and `S3_Source_Engineer-to-Order_Product`

Since interoperability is considered as unidirectional capability of the EISs, two different queries are needed to infer the concepts exchanged between two systems. The first query implements Source process (of the customer) and the second one – Make and Deliver processes (of the supplier). Both queries are using SCOR-KOS OWL ontology to infer about the exchanged entities. The queries consider the flow of SCOR Input/Output elements between the elements of the given process categories.

Hence, information which needs to be sent from the customer's to supplier's EIS and interpreted by the latter can be inferred by using following DL query:

```
(isOutputFrom some (isProcessElementOf value
S3_Source_Engineer-to-Order_Product)) and (isInputFor
some (isProcessElementOf value M3_Engineer-to-Order)) or
(isOutputFrom some (isProcessElementOf value
S3_Source_Engineer-to-Order_Product)) and (isInputFor
some (isProcessElementOf value D3_Deliver_Engineered-to-
Order_Product))
```

The above query results with following SCOR Input-Output elements:

```
Scheduled_Receipts
Inventory_Availability
```

In the opposite direction, following DL query is used:

```
(isOutputFrom some (isProcessElementOf value M3_Engineer-
to-Order)) and (isInputFor some (isProcessElementOf value
S3_Source_Engineer-to-Order_Product)) or
(isOutputFrom some (isProcessElementOf value
D3_Deliver_Engineered-to-Order_Product)) and (isInputFor
some (isProcessElementOf value S3_Source_Engineer-to-
Order_Product))
```

The above query results with following SCOR Input-Output elements:

```
Replenishment_Signal
Production_Schedule
```

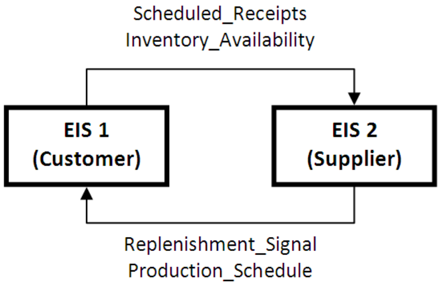

Fig. 3. The example of interoperability requirements

The illustration at Figure 3 shows the data interoperability requirements for two EISs which implement the corresponding SCOR processes, related to the exchange of assets between those. It is very important to emphasize that inferred assets are relevant only when above mentioned SCOR processes environment is considered. In other words, according to SCOR reference, it is sufficient to exchange only the above information between systems to facilitate the customer-supplier collaboration cross-organizational processes, relevant for engineer-to-order manufacturing strategy.

3.1 Explication of the Exchanged Information

In the previous section, the SCOR Input-Output elements which need to be exchanged between the customer's and supplier's EISs in the engineer-to-order collaboration scenario are identified. In order to make the SCOR data interoperable, it is now made explicit by logically relating the resulting SCOR Input-Output elements to the concepts of SCOR-Full ontology. This task is illustrated on the case of one of the exchanged assets – a production schedule.

According to SCORs semantic enrichment - the SCOR-Full ontology, production schedule is considered as sub-concept of "setting" notion and is represented explicitly by the concept "production-schedule", sub-concept of "function-schedule"->"schedule". Thus, the sameness of the instances of "production-schedule" concept of SCOR-Full and "Production_Schedule" instance of SCOR-KOS OWL (of SCOR_Input_Output type) is inferred by the following simple SWRL rule:

```
production-schedule(?x) ⇒ SameAs (?x, Production_Schedule)
```

In SCOR-Full, a setting is defined as a circumstance of any type which affects some course of actions. It is associated with some state or configuration of the tangible (physical-item) or intangible (information-item) resources, namely, with an instance of "configured-item":

```
∀s (setting(s)) ∃ci (configured-item(ci)∧ has-realization(s,ci))
```

The production schedule "setting" is configured by the realization of "production-schedule-item" sub-concept of "information-item". Hence, "production-schedule" concept inherits the anonymous class, defined as (Manchester OWL syntax):

```
has-realization some production-schedule-item
```

"Production-schedule-item" concept inherits anonymous classes, defined as (Manchester OWL syntax):

```
has-product-information exactly 1 product-information
has-production-end-date exactly 1 dateTime
has-production-start-date exactly 1 dateTime
has-product-quantity exactly 1 float
```

where "has-production-end-date" and "has-production-start-date" data properties are sub-properties of "has-date-value" data property, and "has-product-quantity" is sub-property of has-numerical-value data property. "Has-product-information" is a sub-property of "has-realization property". Hence, necessary conditions for having one production schedule item are: 1) to have exactly one product associated; 2) to have a production start date for this product; and 3) to have a production end date for this product.

Similarly, "product-information" information item is configured (hence, its realization is used in the range of first necessary condition above) by having exactly one product id associated:

```
has-product-id exactly 1 string
```

In addition, "function-schedule" concept also inherits the anonymous class:

```
∀fs (function-schedule(fs)) ∃f (function(f)∧ schedules(fs,f))
```

For the concept of "production-schedule", this condition is specialized to:

```
schedules some production
```

As shown above, the SCOR-Full ontology semantically describes the concept of production schedule. This description is mapped to the corresponding instance of the SCOR-KOS OWL model, so it can be used in the context of SCOR processes. When the SCOR-Full ontology is correlated to the EIS (for example, by the logical correspondences between SCOR-Full and EIS's local ontology), then this EIS can be used as facilitator of the SCOR processes in the collaboration environment. Hence, there is no need for specialized software applications.

4 Conclusions

Like other reference models, SCOR framework can be considered as an interoperability tool, since it formalizes the common agreements on the collaboration processes in the supply chain. However, it uses very weak structural formalism to only aggregate enterprise entities in the specific categories and it does this in the context of the supply chain. It does not provide syntactic nor semantic view which can help to express and interpret the model by the computer systems. SCOR-KOS OWL and SCOR-Full aim at closing this gap by providing the semantic model and correspondences of this model with a native one (actually, with the OWL representation of the native model).

In this paper, the above-mentioned OWL models (or ontologies) are used to identify and to make explicit the data interoperability issues related to exchange of the information between the arbitrary systems of customer and supplier in the engineer-to-order scenario. The simplistic process definition of the SCOR model, emphasizes the flow of data throughout the supply chain. This data can be considered as interoperable when its different representations (e.g. in SCOR reference and/or in number of EISs) logically correspond to a single concept of the domain ontology which is a formal description of its meaning. These logical correspondences enable all the systems in a supply chain to correctly interpret all representations of this data. Thus, it becomes possible to make the individual systems, and consequently, processes (SCOR process categories) interoperable, at least within the scope, prescribed by SCOR reference model.

References

1. Stewart, G.: Supply-chain operations reference model (SCOR): the first cross-industry framework for integrated supply-chain management. Logistics Information Management 10(2), 62–67 (1997)
2. Poluha, R.G.: Application of the SCOR model in Supply Chain Management. Cambria Press, New York (2007)
3. Barnett, M.W., Miller, C.J.: Analysis of the virtual enterprise using distributed supply chain modeling and simulation: an application of e-SCOR. In: The Proceedings of the 32nd Conference on Winter Simulation, pp. 352–355 (2000)
4. Chen, D., Vernadat, F.: Standards on enterprise integration and engineering - a state of the art. International Journal of Computer Integrated Manufacturing 17(3), 235–253 (2004)
5. Obrst, L.: Ontologies for semantically interoperable systems. In: 12th International Conference on Information and Knowledge Management. ACM, New York (2003)
6. Berre, A.J., Elvesæter, B., Figay, N., Guglielmina, C.J.: The ATHENA Interoperability Framework. In: The ATHENA Interoperability Framework, Proceedings of the 3rd International Conference on Interoperability for Enterprise Software and Applications (I-ESA 2007), pp. 569–580. Springer (2007)
7. Zdravković, M., Panetto, H., Trajanović, M., Aubrey, A.: An approach for formalising the supply chain operations. Enterprise Information Systems 5(4), 401–421 (2011)
8. Zdravković, M., Trajanović, M., Panetto, H.: Local ontologies for semantic interoperability in supply chain networks. In: 13th International Conference on Enterprise Information Systems. SciTePress (2011)