

SCORVoc: Vocabulary-based Information Integration and Exchange in Supply Networks

Niklas Petersen^{*†}, Irlán Grangel-González^{*}, Gökhan Coskun^{*†}, Sören Auer^{*†},
Marvin Frommhold^{‡§}, Sebastian Tramp[§], Maxime Lefrançois[¶], and Antoine Zimmermann[¶]

^{*}University of Bonn
Bonn, Germany

{petersen, grangel, coskun, auer}@cs.uni-bonn.de

[†]Fraunhofer IAIS
Bonn, Germany

[‡]University of Leipzig
Leipzig, Germany

frommhold@informatik.uni-leipzig.de

[§]eccenca GmbH
Leipzig, Germany
sebastian.tramp@eccenca.com

[¶]École Nationale Supérieure des Mines,
FAYOL-ENSMSE, Laboratoire Hubert Curien,
F-42023 Saint-Étienne, France
{maxime.lefrancois,antoine.zimmermann}@emse.fr

Abstract—Advanced, highly specialized economies require instant, robust and efficient information flows within its value-added and Supply Chain networks. Especially also in the context of the recent Industry 4.0, smart manufacturing or cyber-physical systems initiatives more efficient and effective information exchange in supply networks is of paramount importance. The Supply Chain Operation Reference (SCOR) is a cross-industry approach to lay the groundwork for this goal by defining a conceptual model for Supply Chain related information. Semantics-based approaches could facilitate information flows in supply networks, and enable to analyze, monitor and optimize Supply Chains (in particular for robustness). This paper first reviews existing formalizations of the Supply Chain Council’s SCOR standard. It then introduces the SCORVoc RDFS vocabulary which fully formalizes the latest SCOR standard, while overcoming the identified limitations of existing work. SCORVoc is operationalized by a set of SPARQL queries, that enable to evaluate metrics and key performance indicator (KPIs) defined by SCOR, on-the-fly, in an information systems that adheres to the vocabulary. Finally, we define concrete test scenarios and implement a synthetic benchmark to demonstrate the practicality of SCORVoc.

I. INTRODUCTION

In the past decades, internal enterprise information systems experienced much technical and scientific advancement. However, comparatively little progress was made to improve the exchange of information between enterprises. Until today, most of the communication between enterprises is done via informal channels, such as emails (including file attachments) or telephone calls. Only tier-1 suppliers of major Original Equipment Manufacturers (OEM) are usually fully integrated into the information exchange and corresponding IT-support (e.g. electronic data interchange connections) since these are expensive to deploy and maintain. Informal communication is time-consuming, costly and can become inefficient when crucial information is spread among many different people that each use their own format or system.

This paper focuses on the two following business requirements:

- 1) The production plans of a factory are highly dependent on the incoming supplies, since just-in-time productions aims at keeping the stock as low as possible to reduce *dead capital*. Hence being able to instantly exchange machine-interpretable messages between manufacturer and supplier is critical, for example, in case of supply shortages.
- 2) It is a competitive advantage for a business to be able to assess the reliability of suppliers. However, monitoring the direct suppliers is not enough, since problems deeper in the Supply Chain (delays, strikes, outages, bankruptcies) can have a negative effect even on reliable suppliers. Therefore, it is of paramount importance to be able to pro-actively identify critical suppliers and potential threats in the entire value-added network.

Hence we need a standardized way to represent knowledge about the business processes and the supply network, and to reason with it. This paper focuses on one of the most prominent such standards, namely the *Supply Chain Operations Reference Model (SCOR)*¹ [1]. While SCOR offers a great basis to answer the two identified business requirements, its reference itself only contains textual definitions. Neither machine-interpretable formats for messages, nor automatized ways to reason with such messages exist.

In order to address these limitations, we developed an approach for making the SCOR reference model executable. Our approach comprises the definition of the SCORVoc vocabulary providing an ontological formalization of the terms and concepts defined by SCOR. We argue that using a lightweight RDFS vocabulary is a good step towards applying the SCOR reference in real world applications. The fine-grained Linked Data representation formalism provides a number of

¹<http://www.apics.org/sites/apics-supply-chain-council/frameworks/scor>

benefits for implementing SCOR, namely:

- *Identification* – world-wide unique identifiers facilitate the data exchange and linking in supply networks,
- *Coherence/Reuse* – mixing and mashing of vocabularies as well as schemata enables the reuse and alignment with domain specific formalisms,
- *Granularity* – integration of representation on different levels of granularity,
- *Execution* – query execution in order to automatically aggregate and analyze data,
- *Integration* – bindings to a number of other technology ecosystems comprising XML (with RDF/XML, JSON (with JSON-LD) or HTML (with RDFa).

Thus, this paper addresses the following research question:

How can the SCOR model be formalized and operationalized using the Semantic Web formalisms and the Linked Data principles?

Our approach is structured as follows: Section II first provides a more detailed overview on SCOR together with a discussion about the motivation and the goal of the reference. Then, existing works aimed at formalizing SCOR into a vocabulary (using RDF/S and OWL) are reviewed in Section III.

Finally, Section IV to Section V detail our main contributions:

- In Section IV, we describe our methodical formalization of the terms and concepts. As a result, we defined the SCORVoc vocabulary consisting of 211 classes, 257 properties and 327 instances publicly available on the web. The vocabulary is a first step towards solving the first identified business requirements.
- The SCORVoc vocabulary is partly operationalized using RDF Schema and OWL axioms as well as class constructors. Unfortunately, these languages are not sufficient to capture the semantics of all the metrics and evaluate all the KPIs. In Section IV-C we address this specific need. We introduce 28 queries that adhere to the vocabulary for automatically computing crucial KPIs in a supply network, e.g. with regard to reliability, responsiveness, agility or cost.
- The practical applicability of SCORVoc and its associated SPARQL queries is evaluated in Section V. We demonstrate on generated synthetic benchmark data, that the amount of data and the query execution is easy to handle for partners in supply networks.

Finally, Section VI concludes and discusses future work.

II. OVERVIEW OF SCOR

It is challenging to agree on a standardized way to represent knowledge about the business processes and the supply network. This is partly due to the variety in company size, industry and business models, viewpoints, and granularity of requirements. The *APICS Supply Chain Council*² tackles this



Fig. 1. High-level overview of the Supply Chain Organizations Reference (SCOR v11.0).

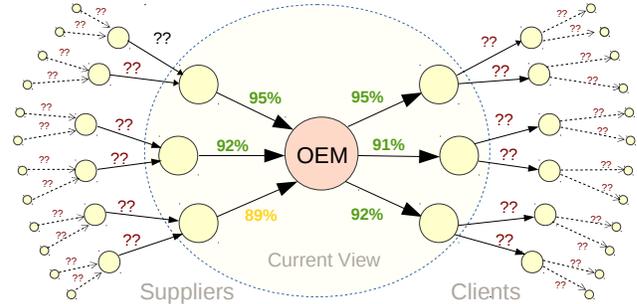


Fig. 2. Supply Chain workflow example: Reliability between Enterprises

challenge, and elaborates a reference model named *Supply Chain Operations Reference Model (SCOR)*³ [1].

The main concept in SCOR is named *process*, and denotes any activity related to the production and logistics. The SCOR model has different conceptualization levels. The Top Level contains the main processes: *Enable, Make, Source, Deliver, Return*. Then, the Configuration Level provides a set of process categories for main processes. Finally, the Process Element Level decomposes the process categories by adding process element definitions and process element information. This leads to a total of 201 definitions of industry-agnostic processes.

By today, SCOR [2]–[5] is in its 11th revision, and has become a mature reference model backed up by many global players (including IBM, HP and SAP). It contains industry-agnostic definitions for 201 processes and 286 metrics. In Figure 1 we give an high-level overview of the reference model.

The motivation behind SCOR is to enable enterprises to diagnose and manage their Supply Chains. Figure 2 illustrates the limited view of an enterprise without any Supply Chain communication. The goal is to extend the view in order to identify poorly performing links and act upon them. Besides communication, it is necessary that each link is measured equally by each partner.

For that purpose, SCOR defines different performance indicators (metrics) including a calculation plan to ensure comparability within the entire Supply Chain. In total, there are 286 metrics which are grouped into five categories: *Reliability,*

²<http://www.apics.org/sites/apics-supply-chain-council/about-apics-scc>

³<http://www.apics.org/sites/apics-supply-chain-council/frameworks/scor>

	Ye [6]	Fayez [7]	Leukel [8]	Sakka [9]	Zdravkovic [10]	Lu [11]	SCORVoc
Reference Version	7.0	7.0	8.0	9.0	n/a	7.0	11.0
Creation Date	2005	2008	2008	2011	2011	2013	2015
Ontology Availability	No	No	No	No	No	No	Yes
Completeness	n/a	Assumed	No	n/a	Assumed	No	Yes
Metric Structure	n/a	Hierarchical	Hierarchical	Hierarchical	n/a	Hierarchical	Queries & Properties
Evaluation using data	No	No	No	No	No	No	Yes

TABLE I
EXISTING EFFORTS FOR FORMALIZING SCOR.

Responsiveness, Agility, Costs and Assets. In Table II, we provide an high-level overview on these metrics. The usage of these metrics allows Supply Chain managers to identify weak and strong links within the Supply Chain.

III. RELATED WORK

Various projects and activities aim at formalizing SCOR into a vocabulary using RDFS and OWL [7]–[11].

The formalization of the SCOR model into an ontology is first addressed in [8]. The authors analyzed the different conceptualization levels of the model and converted them into OWL classes.

In [10], a seminal approach formalizes Supply Chain operations overcoming the semantic weaknesses of the SCOR model. In this work the SCOR-KOS OWL model is presented, which encodes the main entities and properties for SCOR. In addition, the SCOR-Full ontology is a domain ontology for the representation and management of knowledge about Supply Chain operations. The latter presents the core concepts of Supply Chain embedded in SCOR definitions. This effort is also the basis used by Zdravković *et al.* [12] to configure the Supply Chain process. They provide a thread model configuring a specific flow of the Supply Chain studied.

The combination of the SCOR ontology and the *ONTO-PDM*⁴ ontology is addressed in [11]. The ONTO-PDM ontology is used to represent information regarding product development, which is not covered by SCOR. The goal is to create a Supply Chain ontology framework for networked enterprises interoperability.

Sakka *et al.* [9] present a SCOR model as a way to align the business processes with strategical objectives for Supply Chains. Concepts like information and input/output are included to face this alignment. SCOR is modeled using ARIS⁵ thus obtaining a SCOR/ARIS model. Then, XLST transforms the output of SCOR/ARIS into a SCOR OWL ontology.

The work conducted by [14] provides an ontology model to support Supply Chain process modeling and analysis based on the SCOR model. In this work, only part of the SCOR-KOS model [10] related to the definition of input and output entities in SCOR processes is reused.

However, most of these attempts have limitations, summed up in Table I:

- First, these approaches are based on SCOR versions up to 8.0, while the current version is 11.
- Then, the Linked Data principles are not satisfied by these works. They do not reuse existing vocabularies when possible, such as *schema.org* and *Dublin Core*. But above all, vocabularies of these works are not publicly available on the web. We contacted the authors in order to get access to their vocabularies, unsuccessfully.
- Most approaches choose to stay close to the hierarchical structure for processes and metrics given by the original source.
- Finally, none of these vocabularies are ‘operationalized’, and enabled to automatically compute KPIs using data.

The rest of this paper reports on the development and evaluation of the SCORVoc vocabulary, that aims to formalize and operationalize the SCOR model, while alleviating these issues.

IV. FORMALIZATION AND OPERATIONALIZATION

This section describes the formalization of SCOR in a comprehensive RDFS vocabulary, called SCORVoc.

A. Methodology

As seen in the previous section, the main issues in existing formalizations of SCOR is that they have not been developed for a specific version of SCOR, and cannot be continuously updated. On the contrary, we develop SCORVoc using the *VoCol* methodology and support environment [15]⁶ based on the Git version control system. This makes SCORVoc a living artefact, which can be extended and revised by a community of collaborators.

Multiple people were involved in the development of SCORVoc. In order to facilitate the collaborative development, we choose a GitHub repository⁷ for managing vocabulary source files, documentation, queries as well as example data. The Turtle serialization format [16] was chosen due to its simplicity. GitHub’s web interface further provided our domain experts with a very simple way to access the latest SCORVoc version.

We chose the methodology described by Uschold *et al.* [17] for building our vocabulary: i) define the purpose and scope;

⁴ONTO-PDM is an ontology for Product Data Management interoperability within manufacturing process environment, presented in [13]

⁵Business Modeling Approach

⁶<http://eis.iai.uni-bonn.de/Projects/VoCol.html>

⁷<https://github.com/vocol/scor>

Performance indicator	Measures [1]	Result	Example
Reliability	If a task is performed as expected	Percentage	Order Delivered on-time
Responsiveness	The speed in which tasks are performed	# of Days	Average days needed to deliver an order
Agility	The ability to respond to external influences	# of Days	Days needed for an unplanned production increase
Costs	The cost of Supply Chain processes	Amount of money	All labor costs required for a specific product
Asset Management	The ability of efficiently utilize assets	# of Days & Percentage	Inventory days of raw material supply

TABLE II
SCOR PERFORMANCE INDICATOR OVERVIEW

ii) capture the domain knowledge; iii) develop the ontology and integrate it with other existing vocabularies; iv) evaluate it and document it properly.

Purpose and Scope. The purpose of SCORVoc is to provide enterprises with a vocabulary which they can use to express any data related to Supply Chain management (SCM). The users of the vocabulary are therefore enterprises which would like to profit from the benefits of expressing their supply chains in SCOR. The vocabulary aims at being light-weight in order to facilitate its usage for future SCOR compliant IT applications.

Capture. The capture of the domain of interest (Supply Chain data management) was achieved in two ways. First, we used the 976-page SCOR reference [1], with its strong terminological definitions as a major source for studying the domain of interest. Second, we had a domain expert with a deep knowledge (a member of the *APICS Supply Chain Council*⁸) which supported us in the entire process. As a result of many interviews with the domain expert, we acquired a more fundamental understanding of the motivation of SCOR, its strengths but also its weaknesses.

Design. This step is decomposed in the design of the two main aspects of SCOR: processes, described in Section IV-B; and metrics, described in Section IV-C. Existing semantics for each concept and the lack of accessibility of prior approaches to formalize SCOR lead us to create many concepts by ourselves. Nevertheless, various concepts and properties are integrated from well-known vocabularies such as *schema.org*, *SKOS* and *Dublin Core*. We made this decision based on the semantic description of these terms.

Documentation and Evaluation. As illustrated in Section IV-B and Section IV-C, each concept contains a definition together with a full documentation based on the descriptions provided by SCOR. Section V describes the evaluation of SCORVoc.

B. Formalizing Processes

In SCOR, these are the 201 processes. A process represents any business activity between and within enterprises. For most of them, the reference outlines unambiguous text definitions. Since some of them have a rather long name (e.g. *Identify, Prioritize And Aggregate Supply Chain Requirements*), we decided to keep the short name and attach the long version as a label. To stay coherent, all concept and property names follow the camel case notation.

⁸<http://www.apics.org/sites/apics-supply-chain-council>

As proposed in the reference, we created the processes as a hierarchical structure. We defined an abstract super class *Process* with its subclasses *Plan*, *Source*, etc. While certain terms (e.g. *Make*, *Deliver*) do not seem to be self-explanatory, we nevertheless adopted them due to the clear meaning in the SCM domain. All together, the hierarchy consists of three levels (Scope, Configuration, Step). Each level fulfills a certain purpose:

- Level 1 groups processes together,
- Level 2 comprises events, that are to be instantiated in the real world, and
- Level 3 explains in detail how level 2 processes are to be executed (step by step).

Furthermore, the reference defines IDs and clear text definitions for each process. These are reused as is in SCORVoc. Figure 3 visualizes the general structure of the vocabulary with the processes and others main concepts.

Listing 1 shows an example for the full definition of the concept *Enable*. *Enable* is a subclass of the abstract concept `scor:Process`. Each concept contains a definition together with further descriptions provided by SCOR. We further added translations for a variety of languages.

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix : <http://purl.org/eis/vocab/scor#> .
:Enable rdfs:subClassOf :Process ;
rdfs:comment "Enable describes the ...";
rdfs:label "Enable"@en ,
"Permitir"@es ;
skos:notation "E" ;
```

Listing 1. Concept definition example

There are some specificities in SCOR that required the addition of more fined-grained knowledge in the SCORVoc vocabulary.

As an example, SCOR level 3 processes (also called *steps*) have an order of execution of these processes. Therefore, we used the Ordered List Ontology⁹ property `olo:next` to express this relation in our ontology.

SCOR further defined 179 *Practices* to provide a collection of industry-agnostics practices which were recognized for their value. In order to manage talent in the supply chain, concepts such as *Person*, *Skills*, *Experiences*, *Aptitudes* and *Trainings* were introduced. While these concepts are not necessarily needed for the calculation of KPIs, they are all added for sake of completeness.

⁹<http://smiy.sourceforge.net/olo/spec/orderedListontology.html>

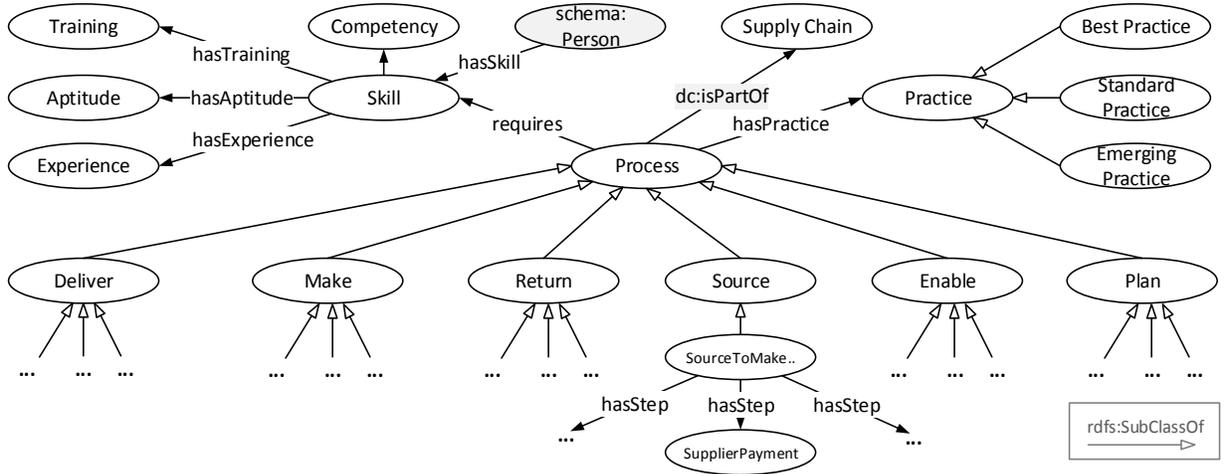


Fig. 3. Overview of the SCORVoc vocabulary (the namespace prefix `schema` refers to `schema.org` and `dc` to Dublin Core).

C. Formalizing Metrics

Metrics are defined by SCOR to evaluate Supply Chains on certain aspects such as reliability or responsiveness.

Similar as the aforementioned processes, the SCOR reference organizes metrics into a hierarchical structure, level 1 being the highest, and level 3 the lowest. SCOR provides for each metrics in a level a calculation plan that takes as an input the value of certain metrics in a lower level.

Let us describe one selected metrics for each category¹⁰:

a) *Reliability*: Level 2 metric *Orders delivered in full* (RL 2.1) of performance indicator *Reliability* measures whether orders are received by the customer in the quantities committed. Its calculation plan is:

$$\frac{\#Orders\ delivered\ in\ full}{\#Orders\ delivered} * 100\%$$

An order is considered as delivered in full once it contains the correct items (RL 3.33, level 3 metric) with the correct quantity (RL 3.35, level 3 metric).

b) *Responsiveness*: Level 1 metric *Order Fulfillment Cycle Time* (RS 1.1) measures the average cycle time in days it requires to achieve customer orders. Its calculation plan is:

$$\frac{\sum Actual\ Cycle\ Times\ for\ All\ Orders\ Delivered}{\#Orders\ Delivered}$$

It depends on multiple other metrics of level 2 such as the time it takes to procure goods and services (RS 2.1), its production time (RS 2.2), the delivery and the delivery retail time (RS 2.3).

c) *Agility*: The metric *Upside Supply Chain Flexibility* (AG 1.1) counts the number of days to achieve an unplanned increase (20% suggested by the reference) in the output of Source, Make and Deliver components.

$$\max(Source, Make, Deliver, SReturn, DReturn)$$

¹⁰For reference, we added SCOR identifiers in parentheses where applicable.

By assuming the production can run concurrently (one strategy provided by SCOR), it requires to identify the process (see metrics AG 2.1-5) within the enterprise whose adaption takes the most time.

d) *Costs*: The metric *Production Cost* (CO 2.004) accounts for all costs involved in the production process.

$$\sum Labor + Automation + Property + Inventory$$

Thus, it depends on metrics which assemble the labor costs (CO 3.014), the automation costs (CO 3.015), the property, plant and equipment costs (CO 3.016) and the governance, risk, compliance, inventory and overhead costs (CO 3.0017).

e) *Assets*: The metric *Cast-to-Cash Cycle Time* (AM 1.1) represents the time it takes for an enterprise to earn money on raw material investments.

$$\sum SalesOutstanding + Inventory - PayableOutstanding$$

Thus, it is necessary to summarize the days between a sale is made and the cash is received (AM 2.1) with the days of sales they were in the inventory (AM 2.2). That sum needs to be subtracted with the days between purchasing raw materials and their actual payment (AM 2.3).

Previous approaches defined a concept for each metric. However in SCORVoc, metrics are represented as data properties, and their calculation plan is represented as an SPARQL query. SCOR metrics are hence 'operationalized', in the sense that all information required for computing the metric is made available in a interoperable way (viz. as Linked Data) and the metrics itself can be translated into queries operating on this information (i.e. SPARQL queries [18]) and returning the respective KPI.

The level 3 metrics are the *data capture* entry point in SCORVoc. They are hence defined as data properties. Their `rdfs:domain` points to their respective processes (given by SCOR) and their range is `xsd:decimal` since they all describe a number between 0-100 (percent values).

Listing 2 shows an example for the full definition of the property `hasMetricRL_33`. Equally as for the documentation of processes, we expressed each property with the definition and the additional information provided by SCOR.

```

:hasMetricRL_33 a owl:DatatypeProperty ;
  rdfs:comment "Percentage of orders ..." ;
  rdfs:label "Delivery Item Accuracy"@en,
    "Exactitud en Entrega de Items"@es;
  skos:notation "RL.3.33" ;
  rdfs:range xsd:decimal ;
  rdfs:domain :ItemAccuracyProcesses .

```

Listing 2. Property definition example

Then, the level 1 and 2 metrics are formalized as SPARQL queries. When triggered, they compute the value of the metric using the values of lower level metrics. Section V contains multiple examples of such SPARQL metric queries.

V. EVALUATION

We evaluate our approach using a qualitative and quantitative method based on the metrics discussed in the previous section. Our qualitative evaluation describes the usage of the SPARQL metrics in a business scenario. In order to do a quantitative evaluation, we developed a SCOR test data generator and measured the execution time of typical queries.

A. Qualitative evaluation

Listing 3 demonstrates a simple example of data expressed using SCORVoc. Since the reference limits itself entirely to processes and performance indicators, we added additional information (using the namespace prefix `ex:`) to make the example more realistic.

The example describes a scenario (`:process_1`) in which goods (viz. keyboards) are received by an enterprise on a certain date. These goods are forwarded to the warehouse which is the reason for the classification of the process as a `:SourceStockedProduct`. Alternatively, if these goods are directly used in the production or for specialized client orders, the process may have been classified as `:SourceMakeToOrderProduct` or `:SourceEngineerToOrderProduct`. This enables enterprises to distinguish more easily between possible unnecessary orders, which end up as *dead capital* in the stock.

As a next step, all information related to this event is captured. `:hasMetricRL_33` represents the accuracy of the items and `:hasMetricRL_50` the quantitative accuracy. Thus, our example shows that this order achieved a quantitative accuracy of 90%.

```

#--- General information -----
:process_1 ex:isSubjectOf "Keyboard X" ;
  ex:hasTimeStamp "2015-10-01T12:52:16" ;
  ex:hasSupplier ex:Logitech ;
  ex:hasCustomer ex:Dell .
#--- SCOR Information -----
:process_1 a :SourceStockedProduct ;
  :hasMetricRL_33 100 ;
  :hasMetricRL_50 90 .

```

Listing 3. Example of data expressed using SCORVoc

Once the Supply Chain related information is captured using SCORVoc, the execution of SPARQL query metrics becomes feasible. As described in Section II, this was the driving force in the development of the SCORVoc vocabulary. Listing 4 shows the *Perfect Order Fulfillment* SPARQL metric. The query compares all complete deliveries (achieving 100%) with all deliveries in total by relying on the respective properties. Applied on the previous example, it returns 0% due to the delivery being incomplete.

The knowledge of these metrics is considered to become a major competitive advantage in the enterprise world.

Besides the previous data example, we will present and briefly discuss how the metrics are realized as SPARQL queries in the following:

Listing 4 presents the *Orders Delivered In Full* metric. Orders are considered to be delivered entirely by SCOR once their item accuracy (RL 33) and quantitative accuracy (RL 50) correspond to 100%. Thus, every order below that value is regarded as incomplete.

```

SELECT ((xsd:decimal(?full) /
  (xsd:decimal(?notFull))
  * 100) as ?result)
WHERE { {SELECT ((count(?deliveredInFull)) as ?full)
  WHERE { ?deliveredInFull :hasMetricRL_33 100 .
    ?deliveredInFull :hasMetricRL_50 100 . }}
  {SELECT ((count(?allDeliveries)) as ?notFull)
  WHERE { ?allDeliveries a :Process . }}

```

Listing 4. Orders Delivery in Full metric

The *Order Fulfillment Cycle Time* is calculated by collecting the respective sum (days) of all source (RS 21), make (RS 22), deliver (RS 23) and deliver retail (RS 24) processes and divides them by amount of all orders.

```

SELECT ((xsd:decimal(?actualTime) /
  (xsd:decimal(?allOrders)) as ?result)
WHERE { {SELECT (SUM(xsd:decimal(?value)) as ?actualTime)
  WHERE { ?order :hasMetricRS_21
    |:hasMetricRS_22
    |:hasMetricRS_23
    |:hasMetricRS_24 ?value . }}
  {SELECT (count(?order) as ?allOrders)
  WHERE { ?order a :Process . }}

```

Listing 5. Order Fulfillment Cycle Time metric

For the calculation of the *Upside Supply Chain Flexibility* metric, it is necessary to gather the value of all flexibility properties (AG1-5) and identify the maximum among them. Similar as a team is only as strong as its weakest part, a Supply Chain is only as agile as its slowest link.

```

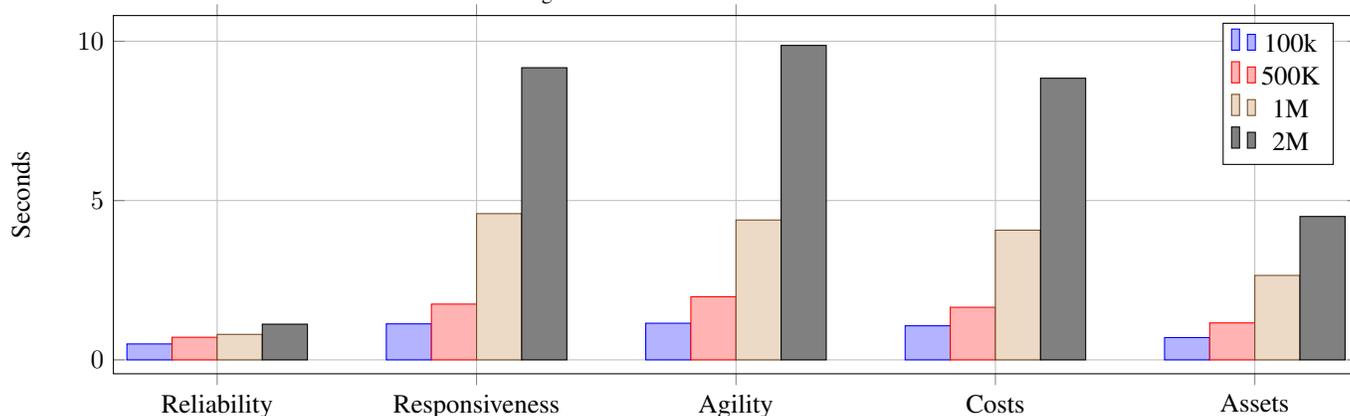
SELECT (MAX(xsd:decimal(?flexibility)) AS ?result)
WHERE { ?order :hasMetricAG_1
  |:hasMetricAG_2
  |:hasMetricAG_3
  |:hasMetricAG_4
  |:hasMetricAG_5 ?flexibility . }

```

Listing 6. Upside Supply Chain Flexibility metric

The *Production Cost* metric (CO 2.004) is dependent on the sum of the metric properties for the costs in Labor (CO 14), Automation (CO 15), Property (CO 16) and Inventory (CO 17).

Fig. 4. SCOR Metric Performance



```

SELECT (SUM(xsd:decimal(?costs)) AS ?result)
WHERE {
  ?order :hasMetricCO_14
        |:hasMetricCO_15
        |:hasMetricCO_16
        |:hasMetricCO_17 ?costs .
}

```

Listing 7. Production Cost metric

Listing 8 presents the *Cast-to-Cash Cycle Time* metric (AM 1.1). The query selects the average time raw materials stays in inventory (AM 2) together with the time the payment is due to by us (AM 1) subtracted by that of our customers (AM 3).

```

SELECT (AVG(xsd:decimal(?inventoryDays))
+ AVG(xsd:decimal(?salesOutstanding))
- AVG(xsd:decimal(?payableOutstanding)) as ?result)
WHERE {
  ?order :hasMetricAM_1 ?salesOutstanding .
  ?order :hasMetricAM_2 ?inventoryDays .
  ?order :hasMetricAM_3 ?payableOutstanding .
}

```

Listing 8. Cash-to-Cash Cycle Time metric

B. Quantitative evaluation

We demonstrate the feasibility of our approach on a SCOR dataset. The only known dataset is *SCORmark*¹¹, which is assembled by a major consulting firm, but is only available for business customers and not open for research. We hence developed a open-source synthetic benchmark data generator, available on GitHub¹².

This generator allows to perform a round-trip between data representation and KPI evaluation. The benchmark allows to assess the performance of an SCORVoc implementation in a systematic and repeatable way. The generator creates data based on a number of parameters: Supply Chain depth, industry and Supply Chain partners. The Supply Chain depth sets the level from one main OEM enterprise to its suppliers' supplier. The industry generates plausible product lines and enterprise names. The Supply Chain partners determine the width of the Supply Chain. A minimum of 2 generates a binary tree to both sides.

Various dataset sizes can be generated in order to assess the scalability as well as the correctness of the metric

Processes	Instances	Related Properties
<i>Source</i>	1.481	28.576
<i>Make</i>	1.785	23.216
<i>Deliver</i>	2.083	22.917
<i>Plan</i>	1.538	18.462

TABLE III
GENERATED DATA OVERVIEW: 100K SCENARIO

SPARQL query implementations. We evaluated the metrics for datasets which contain 100k, 500k, 1M and 2M triples. Table III presents an overview of the generated data for the 100k scenario. While the instances represent different types of processes, the related properties are mostly 3-level data type properties which are required by the metrics (such as `scor:hasMetricRL_50`). The values randomized within a certain range (e.g. >80% for Reliability).

The queries were executed using the *ARQ* SPARQL processor¹³. The machine we used for the experiment contains 8GB of RAM, 256GB SSD and an Intel i7-3537U CPU with 2.00GHz.

Figure 4 summarizes the results of the quantitative evaluation. The *Reliability* and *Assets* queries performed best in our scenario and were also able to be executed fast with datasets larger than 2M triples. The other queries do not scale as well, but still perform with less than 10s query execution time. This is enough for real-world settings. Even for much larger supply networks, we deem query execution performance not to be a bottleneck, since queries are executed relatively infrequently and not by thousands of users. Also, since the number of metrics is limited it is possible to optimize query execution even more, but creating specific indexes or applying caching strategies. Overall this evaluation shows that the approach of having an executable vocabulary is feasible.

Furthermore, the SCOR data generator can be used to systematically assess and evaluate SCORVoc compliant software solutions. Such solutions could be Specific supply network visualizations tools, Supply Chain robustness assessment

¹¹<http://www.apics.org/sites/apics-supply-chain-council/benchmarking>¹²<https://github.com/vocol/scor/generator>¹³<https://jena.apache.org/documentation/query/>

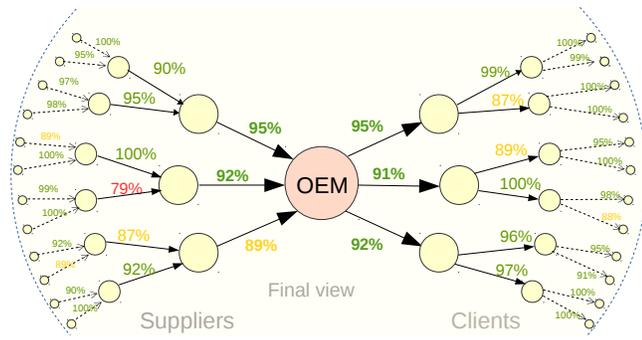


Fig. 5. Final view on the Supply Chain, where KPI information is propagated through the network.

frameworks, or scenario planning tools.

In contrast with Figure 2, Figure 5 represents the full view on the entire Supply Chain.

VI. CONCLUSIONS AND FUTURE WORK

The use of data-centric approaches in engineering, manufacturing and production are currently widely discussed topics (cf. Industry 4.0, smart manufacturing or cyber-physical systems initiatives). The complexity in Supply Chain Management in general is thought to be one of the major bottlenecks of the field. A key issue in engineering, manufacturing and production is to be able to efficiently and effectively manage Supply Chains.

This paper introduces the SCORVoc RDFS vocabulary, that formalizes and operationalizes the SCOR standard. SCORVoc is available on GitHub¹⁴ for collaborative further development and at *purl.org*.¹⁵ We used an innovative methodology to develop the SCORVoc vocabulary, and provided means to automatically compute typical KPIs. Finally, we described comprehensive test scenarios for SCORVoc and implemented a synthetic benchmark. SCORVoc, together with the formalized SPARQL queries, represents a comprehensive approach to facilitate information flows in Supply Chains, and enables the design of SCOR compliant IT applications.

This work is the first step of a larger research and development agenda that aims at providing comprehensive support for information flows accompanying Supply Chains, employing the Linked Data paradigm. The next step is the constitution of a real industry SCOR benchmark, one may test SCORVoc implementations against. One could hence prove the usability of queries, and optimize their execution time.

Finally, let us note we learned from domain experts that SCOR still has a number of limitations. For example a delivery of 9 out of 10 items is described as a 90% success rate. Although for one company this may be an appropriate measurement, for another company, the missing part may actually stop the entire production part. SCORVoc may help identify such limitations, and accompany the improvement of the SCOR specification.

¹⁴<https://github.com/vocol/scor>

¹⁵<http://purl.org/eis/vocab/scor>

VII. ACKNOWLEDGMENTS

This work was partly supported by the following grants from the German Federal Ministry of Education and Research (BMBF) for the LUCID Project¹⁶ (GA no. 01IS14019A) as well as for the the LEDS Project¹⁷ (GA no. 03WKCG11A).

REFERENCES

- [1] S. C. Council, "Supply chain operations reference model," *SCOR Version*, vol. 11, 2012.
- [2] F. Salazar, M. Caro, and J. Cavazos, "Final Review of the Application of the SCOR Model: Supply Chain for Biodiesel Castor-Colombia Case," *Journal of Technology Innovations in Renewable Energy*, vol. 1, no. 1, 2012.
- [3] B. Georgjise, K.-D. Thoben, and F. Marcus Seifert, "Assessing the Existing Performance Measures, & Measurement Systems in Developing Countries: An Ethiopian Study," *Global Journal of Researches In Engineering*, vol. 13, no. 2, 2013.
- [4] F. B. Georgjise, K.-d. Thoben, and M. Seifert, "Implementing the SCOR Model Best Practices for Supply Chain Improvement in Developing Countries," *International Journal of u-and e-Service, Science and Technology*, vol. 6, no. 4, 2013.
- [5] F. Lestari, K. Ismail, A. Hamid, and W. Sutopo, "Designing supply chain analysis tool using SCOR model (case study in palm oil refinery)," in *IEEE Int. Conf. on Industrial Engineering and Engineering Management (IEEM)*. IEEE, 2013.
- [6] Y. Ye, D. Yang, Z. Jiang, and L. Tong, "An ontology-based architecture for implementing semantic integration of supply chain management," *International Journal of Computer Integrated Manufacturing*, vol. 21, no. 1, pp. 1–18, 2008.
- [7] M. Favez, L. Rabelo, and M. Mollaghasemi, "Ontologies for supply chain simulation modeling," in *37th conference on Winter simulation*, 2005.
- [8] J. Leukel and S. Kim, "A supply chain management approach to logistics ontologies in information systems," in *Business Information Systems*. Springer, 2008.
- [9] O. Sakka, P.-A. Millet, and V. Botta-Genoulaz, "An ontological approach for strategic alignment: a supply chain operations reference case study," *International Journal of Computer Integrated Manufacturing*, vol. 24, no. March 2015, 2011.
- [10] M. Zdravković, H. Panetto, M. Trajanović, and A. Aubry, "An approach for formalising the supply chain operations," *Enterprise Information Systems*, vol. 5, no. 4, 2011.
- [11] Y. Lu, H. Panetto, Y. Ni, and X. Gu, "Ontology alignment for networked enterprise information system interoperability in supply chain environment," *International Journal of Computer Integrated Manufacturing*, vol. 26, no. 1-2, pp. 140–151, 2013.
- [12] M. Zdravković, M. Trajanović, H. Panetto, A. Aubry, and M. Lezoche, "Ontology-based supply chain process configuration," in *34th International Conference on Production Engineering, ICPE 2011*, 2011.
- [13] H. Panetto, M. Dassisti, and A. Tursi, "Onto-pdm: Product-driven ontology for product data management interoperability within manufacturing process environment," *Advanced Engineering Informatics*, vol. 26, no. 2, 2012.
- [14] T. Grubic and I. S. Fan, "Integrating process and ontology for supply chain modelling," *International Journal of Computer Integrated Manufacturing*, vol. 24, no. March 2015, pp. 847–863, 2011.
- [15] *VoCol: An Agile Methodology and Environment for Collaborative Vocabulary Development*. Zenodo, Feb. 2015. [Online]. Available: <http://dx.doi.org/10.5281/zenodo.15023>
- [16] D. Beckett, T. Berners-Lee, E. Prudhommeaux, and G. Carothers, "RDF 1.1 Turtle-Terse RDF Triple Language. W3C Recommendation," *World Wide Web Consortium (Feb 2014)*, available at <http://www.w3.org/TR/turtle>, 2014.
- [17] M. Uschold and M. Gruninger, "Ontologies: Principles, methods and applications," *The knowledge engineering review*, vol. 11, no. 02, 1996.
- [18] S. Harris, A. Seaborne, and E. Prudhommeaux, "SPARQL 1.1 query language," *W3C Recommendation*, vol. 21, 2013.

¹⁶<http://lucid-project.org>

¹⁷<http://leds-projekt.de>