

## **SUPPLY CHAIN DYNAMICS IN THE SCOR MODEL - A SIMULATION MODELING APPROACH**

Fredrik Persson  
Christian Bartoll  
Adis Ganovic  
My Lidberg

Matthias Nilsson  
Johan Wibaeus  
Fredrik Winge

Linköping University  
Department of Science and Technology  
Quantitative Logistics Research Group  
Campus Norrköping, 601 74 Norrköping, SWEDEN

### **ABSTRACT**

Supply Chain Simulation (SCS) is today a well-defined branch of discrete-event simulation applications. The differences between different applications are usually small, but in the case of SCS, models tend to be larger, take longer time to build and are harder to validate. To remedy some of these issues in SCS, we propose to use the SCOR model (Supply Chain Operations Reference Model) as a tool to speed up the simulation modeling of supply chains. The SCOR model can be useful in the conceptual phase, the modeling phase, and in the experimental phase of a simulation project. In SCOR Template, a modeling template in Arena, all level 3 processes of Source, Make, and Deliver are modeled to provide the SCS model builder a tool that is fast, follows the SCOR standards in processes and metrics, and simple to use. Here we report on the third version of the SCOR Template.

### **1 INTRODUCTION**

Supply chain analysis with use of simulation modeling is one of many application areas of discrete event simulation. Supply Chain Simulation (SCS) is regarded as a main application area in both research and in practice (Banks *et al.*, 2002). When a simulation model is built from scratch, these projects (in SCS) require several resources and takes a long time. The result, however, provides insights and valuable knowledge of the studied supply chain. A faster and less resource intense method of providing insights and knowledge of a supply chain is to use a process reference model such as SCOR, Supply Chain Operations Reference Model (SCOR, 2012), ARIS (Sheer, 1999), IDEF0 (Vernadat, 1996), or CIMOSA (Vernadat, 1996) in the initial state of the simulation study, cf. Weaver (2010). The drawback is that these modeling approaches are static in nature and does not provide insights and knowledge about system dynamics. Dynamics in a supply chain is something that is difficult to neglect in any supply chain analysis, cf. Towill (1991).

What is needed is then a dynamic tool for supply chain analysis (i.e. SCS) but that uses the structure and definitions of a process reference model. This paper reports on the latest version of the SCOR Template in the simulation software Arena. SCOR Template is an Arena template built on the SCOR Framework to resemble the predefined processes in SCOR. A template in Arena is a set of building blocks (called modules in Arena) that is used when building a simulation model. Previous versions of the SCOR Template has been reported in Persson (2011) and Persson and Araldi (2009). The third version target the third level of SCOR processes instead of the latest two versions that used the second level process definitions.

The Supply Chain Operations Reference Model (SCOR) is a reference model to map, benchmark, and develop the operations of supply chains. Mapping supply chains is done with predefined processes in three levels as referred to previously. For more information on the SCOR model, see SCC (2012) and SCOR (2012). SCOR is a static tool that does not include any dynamic elements. In SCS it is always interesting to measure lead times and lead time variability, delivery accuracy and delivery speed, and to locate bottlenecks. Herrmann *et al.* (2003) reports on the use of the SCOR model combined with discrete event simulation. Their goal was to create supply chain simulation models using the SCOR model and thus, create reusable components in general purpose discrete event simulation software. The e-SCOR software suite from Gensym (2012) is a simulation tool based on the SCOR model. The e-SCOR software works as a building tool for modeling using the SCOR processes and at the same time it allows for a dynamic simulation analysis of the modeled system. Barnett and Miller (2000) reports on the use of e-SCOR in a simple case study. Modeling in e-SCOR is carried out with a drag and drop functionality and allows for user modified building blocks. The e-SCOR simulation model is coordinated with the use of the High Level Architecture (HLA). HLA is a coordination mechanism used to run several independent simulation models as one whole. Coordination is carried out through a common event list for all models being coordinated.

Another SCOR model based simulation tool is the IBM SmartSCOR (Dong *et al.* 2006). The SmartSCOR is a supply chain transformation tool that incorporates both simulation and optimization techniques. Albores *et al.* (2006) present an evaluation of different modeling techniques based on SCOR. In their paper, they model the same supply chain in Witness (a general purpose simulation software), iGrafx process (a business process modeling tool), and in e-SCOR. They find that the use of a SCOR based simulation tool presents an advantage over other more general simulation tools. The advantages can be summarized as the use of standardized processes and metrics, allowing for quick model building and models that are easier to understand (for those already familiar with SCOR). Disadvantages according to Albores *et al.* (2006) with using a SCOR based simulation tool is the lack of flexibility and animation. There is also a risk that reality will be 'altered' to fit the process definitions of SCOR instead of the opposite. Di Dominico *et al.* (2007) present a simulation tool that is based on the Value Chain Operations Reference model (VCOR). In their work they started with a SCOR based tool, but found the scope of SCOR too narrow since it does not include design aspects and a customer focus. Weaver (2010) suggests the use of SCOR as means for building the conceptual model in SCS in order to better comprehend all issues in a supply chain.

Fayez *et al.* (2005), Cope *et al.* (2007), and Cope (2008) present an ontology for automatic simulation modeling of supply chains based on the SCOR model and the IDEF suite of modeling tools. They have created a fast, sharable and easy to used modeling tool for users that have limited knowledge in simulation. Instead, knowledge about the supply chain is fed into the ontology and a simulation model is generated automatically. This provides the user with a stochastic, dynamic and distributed modeling environment to use for supply chain analysis based on the SCOR processes and connected in IDEF3. The model itself is based on a set of predefined modules depicting SCOR processes in level 3, all built in the Arena Template developer function.

It is clear that the SCOR model has a given place in SCS. Previous research, when combining SCOR and SCS, have targeted the level 2 of the SCOR model, cf. Gensym (2012), Persson (2011), and Persson and Araldi (2009), and the level 3, cf. Fayez *et al.* (2005), Cope *et al.* (2007), and Cope (2008). In this version of the SCOR Template, we use the third level of the SCOR model in order to gain more detailed information out from the simulation model. Opposite to Fayez *et al.* (2005), Cope *et al.* (2007), and Cope (2008), we do not allow for automatic generation of the simulation model. Modeling a supply chain is here still considered more of an art than a rigid methodology. Automatic modeling has the drawback of little or no flexibility when creating the simulation model, just like when the level 2 of the SCOR model is used. SCOR Template version 3, based on the level 3 of the SCOR model, provides the same speed and ease to use as other tools, and follows the SCOR process definitions but also provide all the flexibility that the simulation software Arena can provide.

This paper is outlined as follows. First, the SCOR model is presented and the development project of the SCOR Template is reported. After that, the third version of the template is presented, and last, conclusions are drawn on the use of the new SCOR Template.

## 2 THE SCOR MODEL

The SCOR model is a reference model with standardized logistic processes and terminology (Sürie and Wagner, 2008). The SCOR model is made up of three parts, (i) a modeling tool that utilizes standardized processes as building blocks, (ii) a set of key performance measurements (in SCOR; key performance indicators - KPIs), and (iii) a benchmarking tool where companies can compare their KPIs to other companies. The SCOR model is also a methodology for improvements of a company's supply chain. The SCOR model (currently version 10) is being developed by the Supply Chain Council (SCC, 2012). The SCOR model also includes a glossary of common supply chain terminology. Common definitions of key metrics ensure that companies that work with SCOR uses the same definitions and thus, talk the same language. The SCOR model consists of processes in three hierarchical levels. Level 1 – process definitions – consists of the five different process types: *Source*, *Make*, *Deliver*, and *Return* for the information flow and physical flow, and *Plan* to coordinate the other four, see SCOR (2012). The process *Source* contains the procurement of goods and services to meet the demand of raw material, components, and other services needed in production. The *Make* process contains the activities that transform products to a finished state. The *Deliver* process contains the transportation and distribution of finished products to the company's customers. The process *Return* is associated with the returning product to suppliers and receiving returned products from customers. The *Plan* process balances the demand and supply of products and resources in the other processes.

At level 2 – the process type level – more details are introduced in the SCOR model. At level 2, the model distinguishes between make-to-stock (MTS) products, make-to-order (MTO) products, and engineer-to-order (ETO) products. Each level 1 process is divided into subcategories depending on the product. The *Make* process (sM) e.g. is divided into *Make-to-stock* (sM1), *Make-to-order* (sM2), and *Engineer-to-order* (sM3). *Source* and *Deliver* follows the same terminology, with an extra process; *Deliver retail products* (sD4) in the latter. The *Plan* process contains the overall process *Plan supply chain* (sP1) and one planning process for each of the other level 1 processes *Source*, *Make*, *Deliver*, and *Return*. The *Return* process is actually two processes; *Source Return* and *Deliver Return*. These two processes are divided into three sub processes; return of defective products, return of MRO (maintenance, repair and overhaul) products, and return of excess products. Together with all of these level 2 processes, the complete SCOR model also includes the enabling processes. Enabling processes are supporting the other processes and defines most of the methodologies and define planning and control policies. Level 3 – process category – describes the underlying processes of level 2. It is implied by the SCOR model that the level 3 processes are generic for all companies. As an example, the level 2 process *Source stocked product* (sS1) consists of the sub processes; *Schedule product deliveries* (sS1.1), *Receive product* (sS1.2), *Verify product* (sS1.3), *Transfer product* (sS1.4), *Authorize supplier payment* (sS1.5). For further details on the SCOR model, see. SCOR (2012).

The SCOR model defines two types of performance attributes. The customer-facing performance attributes are divided into reliability, responsiveness, and flexibility and the internal-facing attributes are cost and assets. Each of the performance attributes contains a set of metrics. Each metric can be broken down to lower levels (levels 2 and 3) and aggregated from the lower levels up to the top level (level 1). The customer facing metric *Perfect Order* can work as an example. The level 1 metric *RL.1.1 Perfect Order Fulfillment* consists of four metrics in level 2; *RL.2.1 % of Orders Delivered in Full*, *RL.2.2 Delivery Performance to Customer Commit Date*, *RL.2.3 Documentation Accuracy*, and *RL.2.4 Perfect Condition*. The level 2 metric *RL.2.1 % of Orders Delivered in Full* is then in level 3 made up out of *Delivery Item Accuracy* and *Delivery Quantity Accuracy*.

Each process in level 1, 2 and 3 contains a set of metrics that can be used for process evaluation. Each process also contains a list of best practices that can be used in order to improve the metrics. Best prac-

es are the methods and techniques used in the best performing companies in that segment. The SCOR model also provides level 1 metric data to be used for benchmarking purposes (SCC, 2012). With this benchmarking database it is possible to map a company's performance against other companies.

### 3 THE DEVELOPMENT OF SCOR TEMPLATE

The proposed SCOR Template is a template in the simulation software Arena. In SCOR Template version 3, the building blocks are based on SCOR processes in level 3. The previous versions of the SCOR Template were modeled in level 2 of the SCOR model, see Persson (2011), and Persson and Araldi (2009). The main reason for creating the SCOR Template was to create a tool that can be used easily (with a small effort and use of resources and in a short time), can handle modeling with high level of detail, and mimic the dynamic behavior of a supply chain. This was formulated in the four goals of the project, see table 1. In version 1 (Persson and Araldi, 2009), the template followed the SCOR definitions in level 2, and was able to model a supply chain in less than 120 minutes, and cloud handle detailed modeling of processes. In version 2 (Persson, 2011) more processes and metrics were added to the template and less than 60 minutes for modeling. For version 3, as reported in this paper, the SCOR processes modeled in the template followed level 3 of SCOR and the goal is to build a model in less than 20 minutes. The chosen levels of modeling time (duration of the model building activity) does more depict modeling done faster, than a duration of a specific activity. Less than 20 minutes is used to illustrate that modeling is done in a really short time.

Table 1: Project Process

<i>Objective</i>	<i>Version 1 (2009)</i>	<i>Version 2 (2011)</i>	<i>Version 3 (2012)</i>
1. SCOR Compliant	Processes, sS1, sS2, sM1, sM2, sD1, sD2, sP2, sP3, sP4 included	Processes sSR1, sDR1 included. Metrics on Level 1	Processes in level 3 (sS1, sS2, sM1, sM2, sD1, sD2, sP2, sP3, sP4) included.
2. Ease of use	< 120 minutes	< 60 minutes	< 20 minutes
3. Detailed modeling	Yes	Yes	Yes/No
4. Simulation software	ARENA	ARENA	ARENA

Version 1 and 2 of the SCOR Template maps a supply chain in level 2 of the SCOR model. Level 2 differentiates between planning environments such as make to stock (MTS), make to order (MTO), and engineer to order (ETO). The division is rather done based on the positioning of the customer order decoupling point than any other means. The SCOR Template mimics the SCOR processes in level 2 and the template in version 2 includes all processes in the MTS and MTO planning range together with all level 1 metrics. This gives the modeler (the user of the SCOR Template) a great deal of freedom in order to model a supply chain. But, level 2 processes in SCOR are not that flexible and the simulation model implicitly acknowledges that the predefined order of processes in the level 3 mapping of each level 2 SCOR process is correct. This might not always be the case and level 2 processes might be unable to map the reality. That is why SCOR Template version 3 is based on the level 3 SCOR processes instead of level 2.

SCOR Template version 1 and 2 were tested on a total of three cases, cf. Persson (2011) and Persson and Araldi (2009). In Persson (2011) version 1 was tested on cases from the companies Ericsson and Autoliv and in Persson and Araldi (2009) the version 2 was tested on a case from the company Alfa Laval. In all, the cases provided useful information on the usage of the SCOR Template.

### 4 SCOR TEMPLATE VERSION 3 – LEVEL 3 PROCESSES

Version 1 and 2 of the SCOR Template was developed in level 2 of the SCOR model because this was thought to be the most suited level for the simulation of supply chains. Since then, several projects have been undertaken to test the SCOR Template and its modeling ability of supply chains. The idea to shift

the SCOR Template to level 3 has sprung from those test projects where a greater flexibility often was needed than the level 2 can provide. A level 3 SCOR process diagram (thread diagram) includes more information than a level 2 diagram and a level 3 model corresponds closer to a simulation model. Still, the use of SCOR Template, in level 3, is an abstraction at a high meta level compared to a simulation model build from scratch.

#### 4.1 Level 3 Processes

The level 3 processes in Source, Make, and Deliver, the main focus of this paper, are listed in table 2 and 3. Note the division between make to stock and make to order processes. Level 3 sub-processes to the SCOR model depicts a workflow order of processes and is thereby normative in its nature. At the same time, a lot of freedom is given in the application of the SCOR model in level 3 to real company cases. Both the order of sub-processes and the content can be altered to better fit the case. This is one of the main reasons to create the SCOR Template in level 3 instead of level 2 as done previously.

Table 2: Level 3 processes in SCOR - sS1, sS2, sM1, and sM2.

<i>Description</i>	<i>Process name in SCOR</i>	
	Make to stock	Make to order
<b>Source</b>		
Schedule Product Deliveries	sS1.1	sS2.1
Receive Product	sS1.2	sS2.2
Verify Product	sS1.3	sS2.3
Transfer Product	sS1.4	sS2.4
Authorize Supplier Payment	sS1.4	sS2.4
<b>Make</b>		
Schedule Production Activities	sM1.1	sM2.1
Issue Material	sM1.2	sM2.2
Produce and Test	sM1.3	sM2.3
Package	sM1.4	sM2.4
Stage Product	sM1.5	sM2.5
Release Product to Deliver	sM1.6	sM2.6
Waste Disposal (output from sM1.3 and sM2.3)	sM1.7	sM2.7

#### 4.2 SCOR Template version 3

The template structure presented here depicts the modeling of the level 3 sub-processes in SCOR. The template contains the Source, Make and Deliver processes in 27 modules, five Source sub-processes, seven Make sub-processes and 15 Deliver sub-processes. For each sub-process there are a number of inputs and outputs. These were in the development of the modules as far as possible considered so that the defined process should still represent the SCOR processes. In some cases, a number of simplifications were made *e.g.* not every single input was considered since they connect the processes with other processes not yet incorporated.

##### 4.2.1 Make to stock (sS1, sM1, sD1)

In this section, the SCOR Template building blocks, called modules, are presented. For every module, a figure is presented in order to illustrate the inputs and outputs. The vertical lines in the figures indicates the flow of entities, *i.e.* orders or products, and the horizontal lines represents information flows. The Make Plan (MP) in the figures represents the production plan (a forecast) and the Source Plan (SP) represents the raw materials needed to manufacture the forecast.

Table 3: Level 3 processes in SCOR - sD1 and sD2.

<i>Description</i>	<i>Process name in SCOR</i>	
	Make to stock	Make to order
<b>Deliver</b>		
Process Inquiry and Quote	sD1.1	sD2.1
Receive, Enter and Validate Order	sD1.2	sD2.2
Reserve Inventory and Determine Delivery Date	sD1.3	sD2.3
Consolidate Orders	sD1.4	sD2.4
Build Loads	sD1.5	sD2.5
Route Shipment	sD1.6	sD2.6
Select Carriers and Rate Shipments	sD1.7	sD2.7
Receive Product from Source or Make	sD1.8	sD2.8
Pick Product	sD1.9	sD2.9
Pack Product	sD1.10	sD2.10
Load Vehicle and Generate Shipping Docs	sD1.11	sD2.11
Ship Product	sD1.12	sD2.12
Receive and Verify Product by Customer	sD1.13	sD2.13
Install Product	sD1.14	sD2.14
Invoice	sD1.15	sD2.15

In order to schedule and manage the process of delivering products to customer, the sS1.1 module needs to be able to manage plans from both *Make* and *Source*. The MP contains products and the SP includes raw materials from the suppliers needed to manufacture the products. The MP is used for updating the SP, which in our case is executed by an *Assign* module. The Make plan is global and can be accessed by any module, independent of each other. The entities moving through these modules represent raw material, see figure 1.

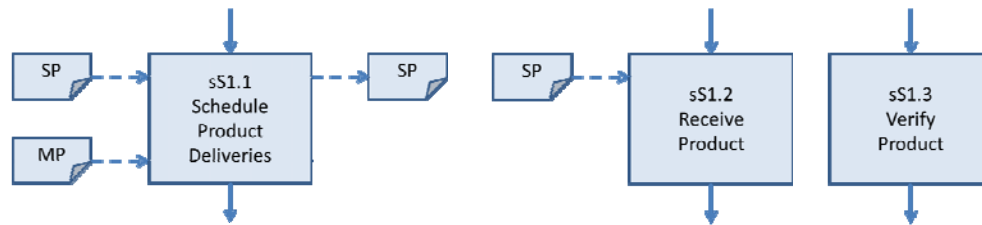


Figure 1: sS1.1 Schedule Product Deliveries, sS1.2 Receive Product, and sS1.3 Verify Product (MP – Make Plan, SP – Source Plan)

The sub-process sS1.2 *Receive product*, see figure 1, is a module that receive raw material from the supplier. The module has a resource that performs the activity and the user has the option to set the time it takes for the activity to run. The amount of received products is predefined in the previous module sS1.1.

In module sS1.3 *Verify Product*, see figure 1, the task is to verify that the received product correspond to the specified requirements and criteria. In this particular case it is decided by a decide module where the user can set the percentage parameter if a product is correct or defect on arrival. Also here is a resource allocated that performs the control of the products and the user can set the control time.

The sS1.4 *Transfer Product*, see figure 2, module represents the raw material storage (MS). In the raw material storage the products are re-packed and held until they are needed. When they are needed there is another resource that transports the raw material to production. Except for the activities there are modules that keep records on the raw material inventory. That information is also kept by global variables and is accessible by every module in the model.

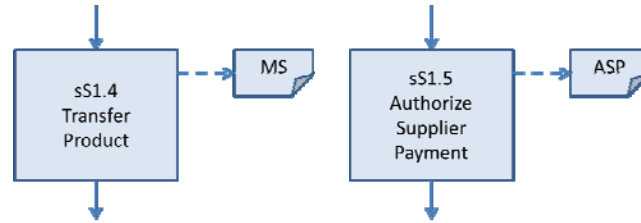


Figure 2: sS1.4 Transfer Product and sS1.5 Authorize Supplier Payment  
(MS – Material Storage, ASP - Authorize Supplier Payment)

The final module of the sS1 process, sS1.5, *Authorize Supplier Payment* is to authorize payments and pay suppliers for the delivered entities. It is therefore necessary that this last module have access to the global matrix of received entities for verification. If the verification is satisfactory, the authorize payment is confirmed, see ASP in figure 2.

The *Make* process includes seven modules which are described below. The entities moving through these seven modules represent both raw material and products. The first module, sM1.1 *Schedule Production Activities*, see figure 3, has the purpose of scheduling the activities, *i.e.* plan sequences depending on surrounding factors and conditions. The production company produces after some kind of forecast plan. In the M1.1 module there is an *Assign module* that access a global variable representing the forecast plan (Make Plan, MP and Source Plan, SP) and schedules production after it.

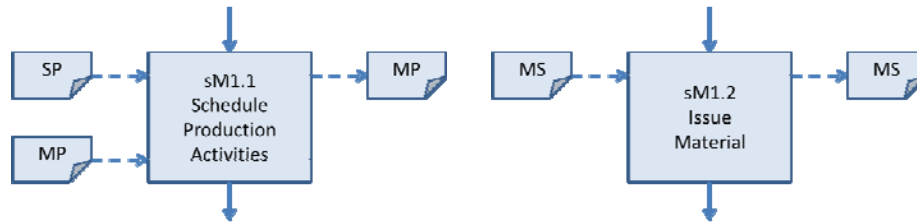


Figure 3: sM1.1 Schedule Production Activities and sM1.2 Issue Material  
(SP – Source Plan, MP – Make Plan, MS – Material Storage)

The next module, sM1.2, *Issue Material*, see figure 3, is needed for placing the managed material at its intended places. Because of this, the module has 3 main tasks: It shall be possible to check if there is any raw material available and if so, send a signal to material storage to release the raw material to production. Finally, it should update the plans for raw materials and production. The material storage (MS) quantity is represented by a global variable, which updates itself every time an entity (raw material) arrives or leaves the storage.

The third module, sM1.3 *Produce and test*, see figure 4, includes the activities of adding value for the products by having the raw material pass through several activities. In the constructed module in this project, these activities have been represented as a resource which represents both production and testing of the products. In the activity of testing the produced products, the user has the possibility of choosing the percentage number of cleared products, *i.e.* how many will pass the test. Those products that do not make the test will be disposed from the system in the module sM1.7 *Waste Disposal* (figure 4). Apart from this, the user also has to set the time of production and testing. From this point, the entities have reached the status of *products*.

The sM1.4 *Package*, see figure 4, is intended to perform the actual packaging of the products. The module include one single resource which represents packaging and other possible activities. The user is intended to set the time for packaging and thereby it is up to the user to make qualified guesses and assumption depending on the number of workers and the time to execute one unit.

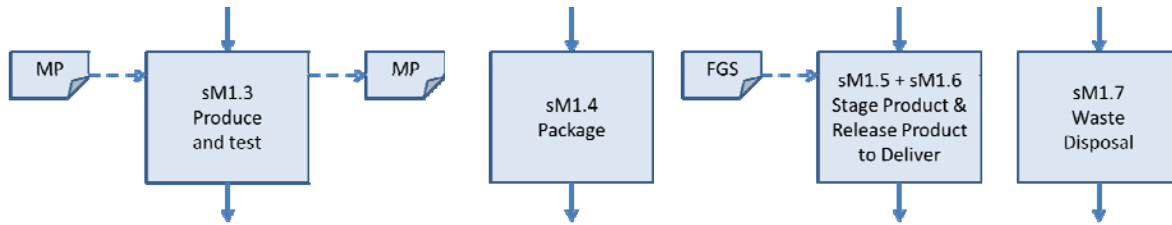


Figure 4: sM1.3 Produce and test, sM1.4 Package, sM1.5 Stage Product, sM1.6 Release Product to Deliver, and sM1.7 Waste Disposal (MP – material Plan, FGS – Finished Goods Storage)

Further the sM1.5 *Stage Product* and sM1.6 *Release Product to Deliver*, see figure 4 for both, has been assembled to one single module sM1.56. The sM1.5 module has the responsibility of transporting the products to temporary holding locations before transportation to a stock for finished goods, Finished Goods Storage (FGS). The second module is responsible for delivering the products to the location for finished goods. Due to the simplified level of complexity for the modules, the authors made the choice of combining the both to one, in order to avoid unnecessary modules. When combined, the module will batch the products and thereafter hold them in a stock (FGS). It will only itself hold the products if the finished goods storage is full, otherwise it will deliver the chosen batch size. In this module, the user has to choose a batch size that is preferred. The authors states that combining these two modules has no effect on the result hence, the user still have the possibility to adjust the same number of parameters.

The last module of the *Make* process is the one mentioned earlier when describing sM1.3, *i.e.* sM1.7 *Waste disposal*. The module has one purpose, which is disposing all waste from production and testing. For that reason, the module is only build up by a *Dispose block* and it does not demand any interaction of the user.

The first module of the *Deliver* process, sD1.1 *Inquiry and Quote*, aims to receive the customer inquiry in order to decide how many products is wanted and use this information together with the forecast plan in order to decide how many products that should be delivered, see figure 5. Since module sD1.1, is the start of the entire chain, a plan containing a forecast of customer demand is created together with all other necessary variables and attributes. In this module orders are created (entities) and then assigned a value representing the size of the order. Every entity is also assigned a unique number, used to keep track of it through the system.

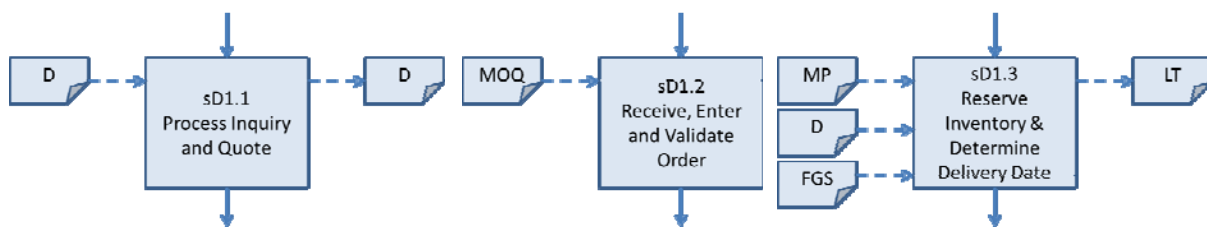


Figure 5: sD1.1 Inquiry and Quote, sD1.2 Receive, Enter and Validate Order, and sD1.3 Reserve Inventory and Determine Delivery Date (D – Demand, MOQ – Minimum Order Quantity, MP – Material Plan, FGS – Finished Goods Storage, LT – Lead Time)

The tasks for module sD1.2 *Receive, Enter and Validate Order*, see figure 5, is to receive, enter and validate incoming orders from customer. The validating part consists of checking that the order corresponds to certain criteria, such as minimum order quantity (MOQ, attribute stated in sD1.1) and price. If the customer order does not fulfill the requirements, the order will be deleted from the system. This may result in an unnecessary amount of finish products in inventory, which can be seen as a miscalculation of forecast.

Module sD1.3 *Reserve Inventory and Determine Delivery Date*, see figure 5, keeps control over customer orders in relation to the company inventory level (FGS) and determines a delivery date. The reser-



vation can only be carried out if the inventory level of finish products (FGS) is greater than customer demand (D). If the inventory level is running low, the reservation will be put on hold until new products have arrived. To determine a delivery date, a lead time (LT) is specified in which time the product must be delivered to customer.

The modules sD1.4 Consolidate order, sD1.5 Build Loads, sD1.6 Route Shipments and sD1.7 Select Carriers and Rate Shipments are all build in the same way. None of the modules have any information as input or output. The modules are taking into account the resources and time needed for execution, see figure 6.

Module sD1.8 *Receive Product from Source or Make*, see figure 6, represent the refill and updating of finish products inventory with the activities receiving and verifying product. When an entity (in this case a product) entering this module, the inventory of finished product (FGS) increases with one. A process, specified with resource and time of execution, illustrates the work of putting away products and recording the right location. A *Hold-module* represent the actually inventory. When an order has been reserved, the amount of products contained in the order, will be subtract from the finished products inventory. The products still physically exist in the inventory but the available amount of products decreases.

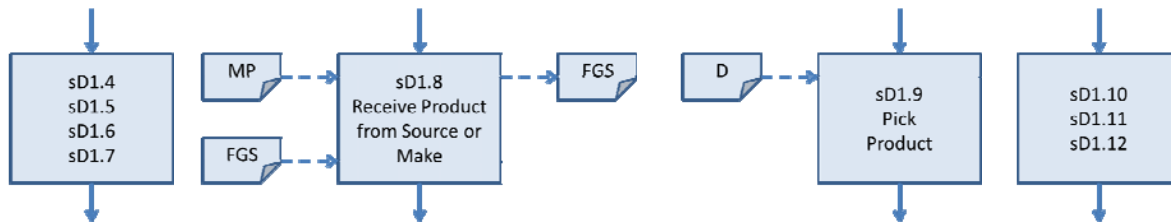


Figure 6: sD1.4 Consolidate order, sD1.5 Build Loads, sD1.6 Route Shipments and sD1.7 Select Carriers and Rate Shipments, sD1.8 Receive Product from Source or Make, sD1.9 Pick Product, sD1.10 Pack Product, sD1.11 Load Vehicle and sD1.12 Ship product (MP – Material Plan, FGS – Finished Goods Storage, D – Demand)

Module sD1.9 *Pick Product*, see figure 6, represents the activity to pick products to a certain order. In this module, two streams of different entity type enter. When an order has reached this point, a signal is declaring that this order shall be picked and products are released from sD1.8. After the entities (products) have been picked (process, resource and time specified), the entities are batched together to one single unit, representing the whole order. The finish product inventory (FGS) is then updated in sD1.8.

Before being shipped, the products need to be packed and loaded in the vehicle. This is represented by the 3 modules sD1.10 *Pack Product*, sD1.11 *Load Vehicle* and sD1.12 *Ship product*. These modules are constructed in the same way, with a single process containing a resource and execution time. None of the modules have any information as input or output, see figure 6.

Module sD1.13 performs *Receive and verify product by customer*. This module represents the work of receiving, verifying that the order was shipped complete and meets the delivery terms. Since the work effort for customer receiving the product do not contribute to the company, sD.13 consist only of one assign updating that the order has arrived in the right way (R).

The sD1.14 *Install product*, see figure 7, module is used when it is necessary to test and install a product at the customer site. The module consists of one *Process module* and one *Assign module*. The process enables use of resource and specifies time of execution. The *Assign module* is used for updating the order status to install (I).

The last module, sD1.15 *Invoice*, initiates the billing process and sends a signal to the financial organization. Depending on the type of company and organization, the billing process can be initialized in different stages of system (P). In this specific case, billing process begins when the products have arrived (R) and are installed (I). To check this, two *Decide modules* have been used, one for each requirement.

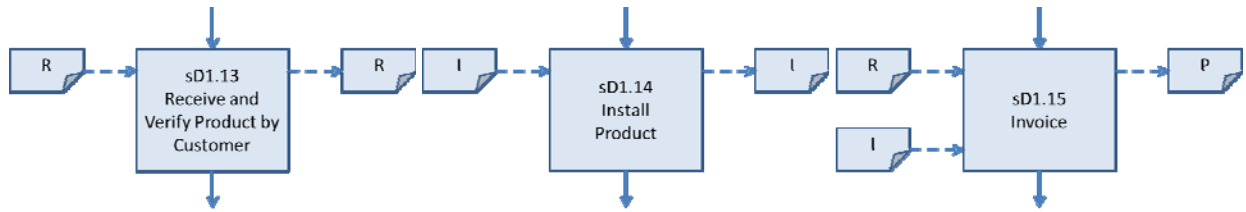


Figure 7: sD1.13 performs *Receive and verify product by customer*, sD1.14 *Install product*, and sD1.15 *Invoice* (R - Receive Product, I - Install Product, P – Pay)

#### 4.2.2 Make to order (sS2, sM2, sD2)

The sS2, sM2, and sD2 processes are modeled in the same way as the make to stock processes as described in the previous section. The main difference is that customer orders drive the planning, not forecasts. Otherwise, the level 2 processes contains the same level 3 processes in SCOR and are modeled in a very similar manner.

#### 4.2.3 Modeling with SCOR Template

The supply chain modeling with the Arena tool is done by dragging and dropping the created modules in the modeling window. To create a flow of entities, which in the simulation represents orders, one must connect the modules to each other. To every module there is a dialog box, which must be filled with data of the specific supply chain set to be simulated, see figure 8. The SCOR Template is created so that only one supply chain can be handled at the same time. If a company has more products more models in Arena is needed.

Figure 8: Data field for sM2.3

### 4.3 Case Study

In order to test the created modules and verify them, a fictitious test case has been created. The case is based on a manufacturing company that produces after a predefined forecast, see figure 9. The forecast is build up by 12 periods with different demand. sD1.1 is then creating 12 orders, one for each period, with a similar normal distribution. In the same way, sM1.1 is starting the production. The difference is that sM1.1 already know how much to produce hence, it is stated in the forecast. The Source module sS1.1 has the function of scheduling the need of raw material with respect to the forecast of how many finished products are needed. For this fictitious case, it takes one entity of raw material in order to produce one

finished product. Also, there is only one product. Resources and time to execute different activities has been specified in each module. A simplification made is that this company only can process one order at the time.

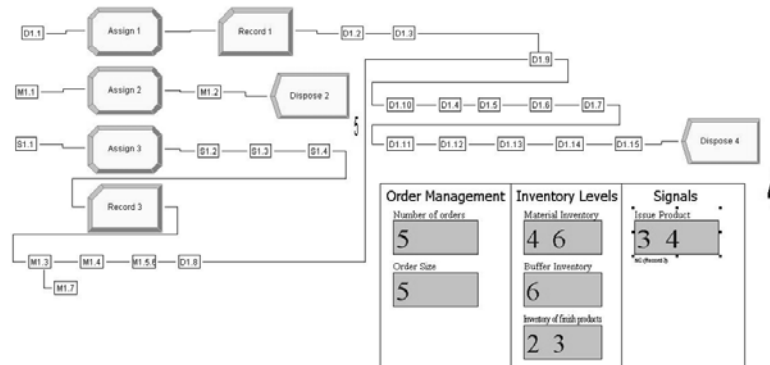


Figure 9: Data field for sM2.3

## 5 CONCLUSIONS

The SCOR Template version 3, with SCOR level 3 processes provides a fast, SCOR compliant, easy to use, Arena template based simulation tool for supply chain analysis. Earlier versions of the SCOR Template have proven useful, although with limited flexibility (Persson 2011 and Persson and Araldi 2009). The contributions of this work is the template like structure for faster model building. The user of a SCOR Template tool can use the tool during SCOR brainstorming meetings, to model a dynamic SCOR mapping of the supply chain at the same time as the supply chain is mapped in basic SCOR processes. The SCOR Template is then a dynamic supplement to a usually static analysis. However, the SCOR Template version 3 requires further development to be really useful, *e.g.* to incorporate the processes *Return* and *Plan*. Still, the use of SCOR Template, for each level 3 process, can save time when analyzing a supply chain. The time to build up a company's supply chain into a simulation model can be reduced significantly compared to modeling every process every time a company should be mapped. Further research will focus on completing the SCOR Template and test the environment on more industrial cases, probably in the construction industry.

## REFERENCES

- Albores, P., Love, D., Weaver, M., Stone, J., and Benton, H., 2006. "An Evaluation of SCOR modelling techniques and tools", *Proceedings of the EuroMOT 2006 Conference*, URL: <http://www.iamot.org/conference/index.php/ocs/9/paper/view/2045/965> , Accessed 2008-01-09.
- Banks, J., Buckley, S., Jain, S., Lendermann, P., and Manivannan, M. 2002. "Panel session: Opportunities for simulation in supply chain management", *Proceedings from the 2002 Winter Simulation Conference*, pp. 1652-1658.
- Barnett, M. W. and Miller, C. J., 2000. "Analysis of the Virtual Enterprise Using Distributed Supply Chain Modeling and Simulation: An Application of e-SCOR", *Proceedings of the 2000 Winter Simulation Conference*, pp.352-355.
- Cope, D. 2008. *Automatic Generation of Supply Chain Simulation Models from SCOR Based Ontologies*, Doctoral Dissertation, Department of Industrial Engineering and Management Systems, University of central Florida, Orlando, Florida, USA.
- Cope, D. Fayez, M. S. Mollaghasemi, M. Kaylani, A. 2007. "Supply Chain Simulation Modeling Made Easy: An Innovative Approach", In *Proceedings of the 2007 Winter Simulation Conference*, pp. 1887 – 1896.

- Di Domenico, C., Ouzrout, Y., Savinno, M.M., and Bouras, A. 2007. "Supply Chain Management Analysis: A Simulation Approach of the Value Chain Operations Reference Model (VCOR)", in *Advances in Production Management Systems* edited by Olhager, J. and Persson, F., Springer.
- Dong, J., Ding, H., Ren, C., and Wang, W. 2006. "IBM SmartSCOR – A SCOR Based Supply Chain transformation Platform Through Simulation and Optimization Techniques", In *Proceedings of the 2006 Winter Simulation Conference* edited by Perrone, Wieland, Liu, Lawson, Nicol, and Fujimoto, pp. 650-659.
- Fayez, M. Rabelo, L. and Mollaghasemi, M. 2005. "Ontologies for Supply Chain Simulation Modeling", In *Proceedings of the 2007 Winter Simulation Conference* edited by Henderson, Biller, Hsieh, Shortle, Tew and Barton, pp. 2364 – 2370
- Gensym, 2008. URL:<http://www.gensym.com>, Accessed 2008-01-09.
- Herrmann, J. W., Lin, E. and Pundoor, G. 2003. "Supply Chain Simulation Modeling using the Supply Chain Operations Reference Model", In *Proceedings of the ASME 2003 Design Engineering Technical Conference and Computers and Information in Engineering Conference*, Chicago, Illinois USA, September 2-6.
- Persson, F. 2011. "SCOR Template - A Simulation Based Dynamic Supply Chain Analysis Tool", *International Journal of Production Economics*, Vol. 131, Issue 1, pp. 288 – 296.
- Persson, F. and Araldi, M., 2009. "The Development of a Dynamic Supply Chain Analysis Tool - Integration of SCOR and Discrete Event Simulation", *International Journal of Production Economics*, Vol 121, Issue 2, pp. 574–583
- SCC, 2012. URL: <http://www.supply-chain.org>, Accessed 2012-01-03.
- Scheer, A-W. 1999. *ARIS: Business Process Framework*, 3<sup>rd</sup> ed., Springer, Berlin.
- SCOR, 2012. *SCOR Version 10.0 Overview*, URL: <http://www.supply-chain.org>, Accessed 2012-01-10.
- Sürle, C. and Wagner, M., 2008. "Supply Chain Analysis", in *Supply Chain Management and Advanced Planning* edited by Stadtler H. and Kilger, C., 4th Ed., Springer-Verlag, Berlin, pp. 38-63.
- Towill D.R., 1991. "Supply Chain Dynamics", *International Journal of Computer Integrated Manufacturing* 4(4), pp. 197-208.
- Vernadat, F.B. 1996. *Enterprise modeling and integration: principles and applications*, Chapman & Hall, London.
- Weaver, M. 2010. *A Simulation Conceptual Modelling Methodology for Supply Chain Management Application*, Doctoral Dissertation, Aston University, UK.

## AUTHOR BIOGRAPHIES

**FREDRIK PERSSON** is Associate Professor in Quantitative Logistics at the Department of Science and Technology at Linköping University, Sweden. His research interest includes modelling and simulation of manufacturing system and supply chains, especially in the construction industry. Of special interest are simulation methodology and other modelling approaches applied to supply chains. He is currently head of the Quantitative Logistics group in Brains and Bricks – a research consortium focusing on construction industry related topics. His email address is [fredrik.persson@liu.se](mailto:fredrik.persson@liu.se).

**CHRISTIAN BARTOLL, ADIS GANOVIC, MY LIDBERG, MATTHIAS NILSSON, JOHAN WIBAEUS, and FREDRIK WINGE** are undergraduate students in the Master program Communications- and Transport Systems at Linköping University. In the Master program it is possible to focus on logistics and simulation.