Performance and Cost Analysis of Supply Chain Models*

F. Bause, M. Fischer, P. Kemper and M. Völker

Dept. of Computer Science IV,
University of Dortmund,
44221 Dortmund, Germany
{bause,fischer,kemper,voelker}@ls4.cs.uni-dortmund.de

Abstract

In this paper we introduce a general framework for the modeling, analysis and costing of logistic networks including supply chains (SCs).

The employed modeling notation, the so-called *Process Chain* paradigm, is specifically developed for the application field of logistic networks which includes SCs. We view SCs as discrete event dynamic systems (DEDS) and apply corresponding simulative techniques in order to derive performance measures of the *Process Chain* model under investigation. For this purpose *Process Chain* models are automatically transformed into the input language of the simulation tool HIT. Subsequently, a cost accounting model using the performance measures is applied to obtain costs which are actually subject of interest.

The usefulness and applicability of the approach is illustrated by a typical supply chain example. We investigate the impact of an additional SC channel between a manufacturer and web-consumers on the overall supply chain costs.

1 Supply Chain Modeling, Analysis and Management

The analysis and evaluation of system performance has become an important and demanding field in the last decades. In spite of the existence of numerous modeling and analysis frameworks, system evaluation is still a challenging task, because systems become increasingly complex and large and therefore performance is increasingly difficult to assess. Model-based analysis not only aims at determining whether or not a system meets certain objectives like performance and costs, but also understanding how system performance can be improved, e.g. by elimination of bottle-necks.

In this contribution we focus on systems describing supply chains (SCs). The SC management (SCM) may be considered as a management approach for designing, implementing and evaluating the flow of materials, products and information among multiple participants. These act as suppliers, manufacturers, wholesalers and retailers which are all involved in procuring, producing and delivering products.

SCs show an increasingly complex behavior reflected by trends towards the sale of goods via the internet and outsourcing. Last but not least competitive pressure, falling profit margins and customized demands make business life challenging and drive enterprises to use information technology to support the design, implementation, monitoring and controlling of SCs. Executives in the SCM field are searching for ways to cope with the complexity of the task and there is a need of a concept resp. modeling and analysis framework according to [12].

---

*This work is supported by DFG, collaborative research center 559 'Modeling of Large Logistic Networks'.

Paradigms of modern business have changed, enterprises do not compete as autonomous entities, but rather as SC processes [12] including their logistic networks. One consequence is that thinking in terms of processes rather than business units has achieved attention and acceptance in the SCM field.

Various notations have been developed for modeling and analysis of SCs for the purpose of

1. planning, design, implementation, maintenance (operational level) and
2. quantification of certain performance measures like costs (performance level).

Both areas interdepent and therefore, although we mainly focus on performance quantification, we give a comprehensive overview about the operational level. Here the intended use of models concerns the (re)-design and the (re)-configuration of (existing) SCs aiming at best practice implementations. Insofar modeling at this stage helps executives to make strategic decisions and addresses Demand Planning 'DP' (which product to offer), Supply Network Planning 'SNP' (where and how to produce) and Product Planning/Detailed Scheduling 'PP/DS' (required resources). Graphical representations of SC models can help executives to find intuitively best practice implementations. Supplementary, models can be analyzed in an automatic fashion by consistency checks and by optimization techniques based on underlying integer programming models.

We view the *Supply Chain Operations Reference* (SCOR) model [14] as a helpful guidance in this area, because it provides a standardized way of modeling SCs and a predefined set of basic business activities associated with all phases of SC processes. Particular SCs are described by user-specified configurations of 26 predefined, so-called core *process categories*, such as `Source Make-to-Order Product` (S2) and `Deliver Make-to-Order Product` (D2). Figure 1 shows a typical SC scenario. Arrows indicate process categories which are labeled with their short names. Such predefined process categories encompass every effort involved in 1) the receipt of order, 2) the supply for sub-assemblies and the treatment of materials via procurement channels between supplier(s) and manufacturer, 3) producing, stocking and routing the shipment of purchased products and 4) delivering final products to the customer to fulfill any requirements of the contract. Therefore, to set up a SCOR model at the configuration level, knowledge about how the SC is configured, i.e., who are the members (suppliers, manufacturer etc) and what are their own parts of the whole SC process, is required.

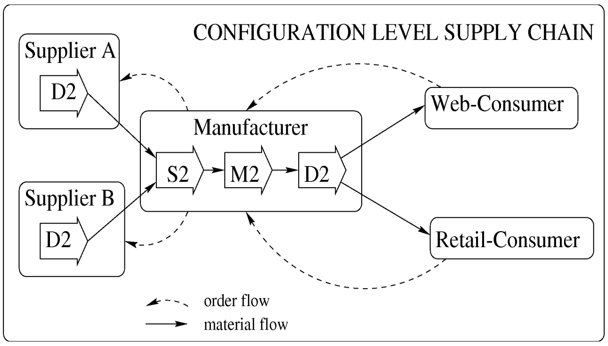

Figure 1: Configuration level (in accordance to SCOR terminology) of SC scenario

Numerous commercial software for planning SCs is at hand and we name only few. The market leader SAP offers the APO (Advanced Planning and Optimization) package [3] which consists of components for 'DP', 'SNP' as well as for 'PP/DS'. e-SCOR [2], developed by Gensym and based on the SCOR standard, enables the modeler to simulate and evaluate alternative SC configurations.

Henceforth we consider the second area, the quantitative resp. performance level. In contrast to SC planning, performance analysis of SCs presumes the existence of a particular SC-configuration including the definition of SC members and their interconnection pattern. Furthermore a detailed specification of activities, supplemented by timing information and assigned resources which are requested for their execution, is required in advance. Insofar models and their specification languages become more formalized at this stage of description. During the introduction of our proposed modeling language in section 2 we will return to this point.

As mentioned above various notations have been developed for modeling and (performance) analysis, see for example [9] for a recent overview. Most of them belong to the class of formalism for discrete event dynamic systems (DEDS) [6]. DEDS consider systems which are characterized by discrete states and atomic state changes among states. This is quite natural in a business context, e.g. in bookkeeping where quantities are discrete (money, stock of materials) and entries are made completely or not. We take this point of view as well and consider SCs as a DEDS in which the current state of affairs changes due to the occurrence of an event like acceptance of an order or delivery of a product at the customer etc.. Therefore high-level notations describing DEDS are of particular interest here. We believe, that modeling and analysis frameworks well-known in the performance analysis of computer and communication networks can be partly adopted for the purpose of SC modeling and analysis. In the following we will discuss this issue.

2 Modeling Supply Chains using the Process Chain Notation

A modeling language which is especially designed to the needs of logistic networks including SCs is the *Process Chain* notation. Originally it has been primarily developed for the planning resp. design of SCs, see [11], and it has been successfully applied in consulting projects and by this means proved its value. A subsequently introduced more formalized definition [8] allows to combine modeling and analysis with respect to performance evaluation and goes beyond a modeling and analysis framework for the design of SC. Current work is dedicated to upgrade modeling and analysis capabilities in order to allow not only performance evaluation, but also to quantify models with respect to costs and ecological aspects.

The *Process Chain* notation is the joint specification language in a large, interdisciplinary, and longterm collaborative research center (SFB 559) at the university of Dortmund [1]. The approach has been implemented in a toolkit which provides a graphical user interface for model generation and integrates several analysis engines for qualitative and quantitative assessment of *Process Chain* models.

The *Process Chain* notation follows a classical point of view obtained in performance analysis: a workload specification characterizes what tasks need to be done, a machine specification describes the resources that can cope with certain tasks, and finally these two specifications need to be matched within a single model to clarify which (sub-)task is performed by which resource. Clearly, processes have strong similarities with workload specifications. A complex business process consists of many subtasks, some of them may require a specific order, others may proceed in parallel or trigger other, additional, and independent processes. A successful strategy to cope with large and complex models is to enforce a hierarchical structure. For processes it is natural to build a hierarchy based on refinement, such that a complex task within a process is only referenced and its detailed description is given in an additional submodel. This is defined in a recursive manner to achieve self-similarity and a homogeneous design: a process consists of a set of tasks which can be processes themselves. From the machine/resource point of view, companies are hierarchically structured into business units, departments etc. This type of hierarchy is build on containment, each unit contains an amount of smaller units and resources as its own and offers a set of services to other, external units. The *Process Chain* notation integrates both types of hierarchies into a single one: a unit (functional entity) offers a set of services described by processes (chains) to its environment and owns a set of resources to cope with the demand for services.

The *Process Chain* notation provides a variety of modeling constructs with regard to 1) the workload characterization, 2) resources, 3) behavior of

---

[1] `http://www.sfb559.uni-dortmund.de/eng/index.htm`

individual processes in terms of their sequence of activities and 4) fork-join mechanisms for parallelism and synchronization between processes.

The main structuring elements are *Functional Entities* (FEs), that encapsulate one or more *Chains*. A chain can be viewed as a structured and measured set of activities starting with *Sources* for process creations (demand), denoted by a circle with midpoint ⊙, followed by a chronological sequence of *Process Chain Elements* (PCEs) describing activities, denoted by arrow-like hexagons, and the chain is accomplished by a sink, denoted by ⊗. Figure 2 depicts a PCE and its labeling and figure 7 shows an entire model. Graphical elements are ordered from left to right. The horizontal connections between PCEs indicate a sequential behavioral pattern of processes. Branches into and merges from alternative sub-chains are allowed and represented by vertical bars. As mentioned above, *Process Chains* are hierarchically structured. PCEs may invoke sub-chains from so-called *subordinated FEs*. Subordinated FEs and sub-chains are described in the same manner as their super-ordinated FEs and chains (self-similarity). Actually, two types of subordinated FE are distinguished: user-defined FEs and simple, predefined FEs of type *Server* or *Counter*. Servers capture timing aspects and model active, possibly shared resources, i.e. machines, assembly lines, workers. In principle, servers correspond to single stations in queueing networks. Counters model space and describe passive resources, i.e. stores and waiting areas of usually restricted capacity. Figure 2 shows a PCE and its labeling.

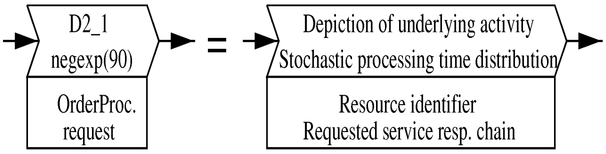

Figure 2: Labeling of Process Chain Elements

Performance analysis requires to incorporate timing information into the model. All inter-arrival times of customer orders, as well as all delays due to processing and transportation are described by random variables. For example in figure 2 the processing time is specified by a random variable with a negative exponential distribution. In fact we distinguish between two types of time delays arising in SCs. On the one hand there are delays of predictable nature like periods stipulated for delivery. In certain circumstances such delays are depending on states captured by the model, i.e. load situations, or such delays are influenced by some stochastics. However, delays are known in advance and may be viewed as fixed model input parameters. On the other hand, there are also delays of unpredictable nature due to congestion at shared resources, e.g. caused by servers or if other required resources are not readily available, and their quantification requires performance analysis.

3 Model Translation, Simulative Analysis and Costing

The formal modeling paradigm described above offers the opportunity to analyze process chain models with respect to performance. Because of formalization, the semantics is made explicit and thus the calculation of performance figures can in principal be done automatically. A drawback of any such newly introduced modeling formalism is that it does not automatically participate in state-of-the-art performance analysis techniques like broadly used simulation or analytical methods for which several tools are available, see [5] for an overview.

We bridge the gap between the domain-specific *Process Chain* modeling notation and analysis capabilities by providing automatic model translations into the input language of the discrete event simulator HIT [4]. Many systems are too complex or do not fulfill restrictions imposed by analytical solutions. But in certain circumstances, at least submodels are analytically tractable and by this means analytical methods are not generally excluded and can be used to compute aggregates representing those submodels in exact or approximated fashion. However, for this reasons we mention the existence of translations from well-defined subclasses of *Process Chain* models also into queueing networks or Petri nets [13], see also figure 3. But in this contribution we only employ the transformation giving a simulation model for HIT.

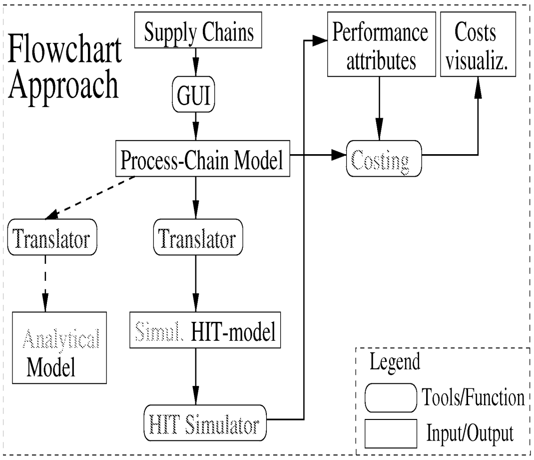

Figure 3: Steps towards performance attributes and cost key figures.

Performance analysis gives results on technical properties like throughput or utilization. As suggested in [7], these technical measures can be used to derive cost key figures supporting several costing models.

For example, if we assume that $\lambda(r,a)$ gives the throughput at a service center resp. resource labeled as r invoked by activity a and $PC^*_c(r)$ gives the costs incurred by using this resource, hence the product $\lambda(r,a) \cdot PC^*_c(r)$ gives the expenses per time unit. Similarly, lead times $t(r,a)$ and populations $n(r)$ can be used to compute costs. By this means costs are calculated ex-post from performance results. In the following example we measure costs in Euro[2] (€).

For a particular resource r we distinguish between 3 types of costs which are considered to be (input) parameters of the costing model: 1) fixed costs $\mathrm{FC}^*_t(r)$ [€ per hour], 2) operational costs $\mathrm{OC}^*_t(r)$ [€ per item and hour] and 3) performance costs $\mathrm{PC}^*_c(r)$ [€ per call] resp. $\mathrm{PC}^*_t(r)$ [€ per time (hour)]. All parameters are labeled by an asterisk * in order to guard against misunderstandings and to distinguish between given input parameters of the costing model ($\mathrm{FC}^*_t, \mathrm{OC}^*_t, \mathrm{PC}^*_c, \mathrm{PC}^*_t$) and computed costs (no asterisk). Additionally, costs are subscripted by $_t$ and $_c$ to clarify the measuring unit. In the current implementation, cost parameters cannot depend on their origins (activities and processes).

Similar to the parameters listed above, computed costs are divided into: 1) fixed costs $\mathrm{FC}_t(r)$, 2) operational costs $\mathrm{OC}_t(r)$ and 3) performance costs $\mathrm{PC}_{c/t}(r,a)$ for a particular resource r called resp. consumed by activity a. For fixed costs and operational costs origins cannot be determined and therefore costs are not tracked to a certain activities a.

Indirect costs increased in the last decades [1] and therefore costing systems like 'activity based' (ABC) or 'process chain costing' in general that capture costs at lower levels (activities) and therefore allow to allocate indirect costs fairly, attracted attention in the last years. Hence, costs can be factored out and assigned to processes or even to certain activities involved in a particular process. ABC is very related to what we refer to as performance costs. ABC can roughly subdivided into two models, see figure 4. These models define logical and quantifiable relationships between resources, activities that consume resources and cost objects that consume in turn activities. The so-called **resource cost driver model**, see figure 4, assigns resources to activities. In our graphical notation its counterpart is given by boxes attached below the PCEs. The so-called **activity cost driver model** describes how activities are consumed by costs objects and which expenses are incurred. Because of we face dynamic models, quantified measures of how cost objects consume activities in terms of lead times and frequencies are not known in advance. Therefore results of performance analysis must be exploited. In particular we need performance mea-

---

[2] The Euro (€) is the official currency of the European Monetary Union since January 1999 and the changeover will be on January 2002. For comparison only: €1=1168 Won=110 Yen.

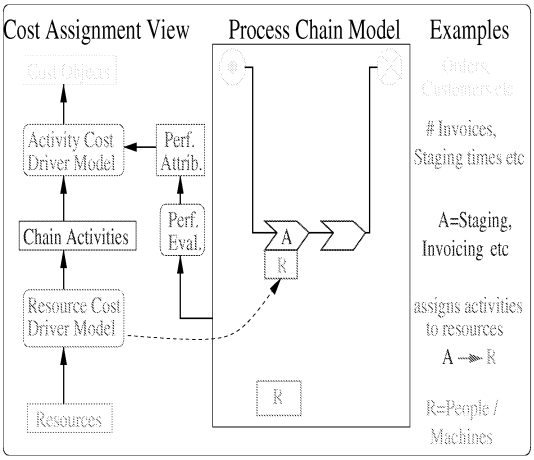

Figure 4: Activity based costing and its relationship to key cost figures derived from *Process Chain* notation.

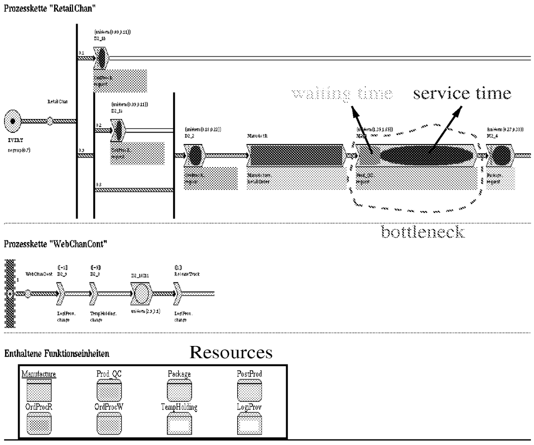

Figure 5: Performance attributes and their visualization used for bottle-neck detection.

sures with respect to certain activities, or more loosely speaking with respect to origins. The performance evaluation tool HIT offers this possibility by defining paths in the calling hierarchy of used and provided resources. Performance measures can then be assigned to processes having followed the specified path. E.g. for the model in figures 7 and 8 HIT is able to determine the throughput at process `RetailOrder` in the subordinated FE `Manufacturer` for for all processes belonging the process class `RetailChan`, i.e., are invoked from process `RetailChan`. Because we actually strive for costs, performance results must be supplemented by costs information, for example costs per invoking an activity resp. resource and costs related to time-intervals elapsed until a cost object passed a particular activity. This approach has been automated [10] so that the data is available to fill the above mentioned costing systems. Automatically generated HTML-reports support information spreading and impressive and easy-to-understand visualization of performance and cost values. Figure 5 gives a flavor of how visualized performance attributes appear in HTML reports.

Finally, important relationships between the input parameters of the costing model and technical performance attributes on one hand and derived cost figures on the other hand are listed in the following equations. Notice, fixed cost $\mathrm{FC}_t(r)$ are load independent and therefore they do not depend on technical performance attributes and equal to their corresponding costing model parameters, see equation 4.

$$\begin{aligned}
\mathrm{PC}_c(r,a) &= \mathrm{PC}_c^*(r) + \mathrm{PC}_t^*(r) \cdot t(r) && (1)\\
\mathrm{PC}_t(r,a) &= \mathrm{PC}_c(r,a) \cdot \lambda(r,a) && (2)\\
\mathrm{OC}_t(r) &= \mathrm{OC}_t^*(r) \cdot n(r) && (3)\\
\mathrm{FC}_t(r) &= \mathrm{FC}_t^*(r) && (4)
\end{aligned}$$

Aggregated performance costs associated with complete SCs are of primary interest. For this reason we introduce the notations PC(`WebChan`) and PC(`RetailChan`) describing the overall performance costs incurred by all order processing activities. These costs are calculated as the sum of costs of particular activities involved in the SC under consideration.

In the following we will present performance results and ex-post derived costs for our running example.

4 Example

The objective of model-based quantitative analysis is to capture certain attributes of involved SC processes including their performance and costs.

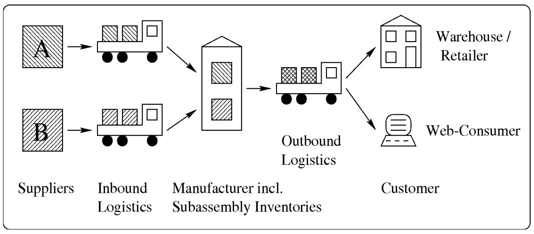

Figure 6: SC scenario under consideration

We consider two interacting SCs labeled as `RetailChan` and `WebChan`, see figures 6 and 1. The `Web-Consumer` SC is assumed to gain importance from the erosion of sales from the `Retail-Consumer` SC, i.e., successful EC applications impose a drain of demand from Retail-Consumer towards Web-Consumers. Latter produce a larger number of smaller orders, we assume that Retail-orders have five-fold size. We consider several scenarios (Scen) and vary the arrival rates λ_W [λ_R] of Web-Consumers [Retail-Consumer]. So we vary the percentage of order volumes perc of Web-Consumers, see table 1. The overall demand remains unchanged. All order interarrival times are assumed to be exponential, i.e., we model Poisson arrival streams [5].

Scen	λ_W	λ_R	perc
1	0.50	1.00	9.01
2	1.25	0.85	22.73
3	2.00	0.70	36.4

Table 1: Load imposed by Web-Consumers and Retail-Consumers

However, both chains have a similar configuration and differ only from how the customer contact is organized and how resource consumptions resp. allocations are quantified. Both SCs consist of the same two suppliers labeled as 'A' and 'B', a single original equipment manufacturer which offers a particular product and logistic service providers which realize the out-bound transport, see figures 1 and 6. To simplify matters we assume, that the manufacturer offers exactly one product 'AB' which is assembled from two sub-assemblies of type 'A' and 'B'. Retail-Consumer and Web-Consumers order product 'AB' with a certain, consumer-specific purchase order quantity and frequency. We incorporate a simple warehousing policy for sub-assemblies. The manufacturer procures sub-assemblies from the suppliers with a certain quantity M if and only if the stock of inventory has been decreased by M. The value of M exceeds the mean customer purchase order quantity and therefore procurement is not required upon each receipt of customer orders.

The activities inside the manufacturer are modeled under consideration of predefined execution processes from the SCOR model. Figure 1 indicates the processes we use: `Source Make-to-Order Product` (S2), `Make-to-Order` (M2) and `Deliver Make-to-Order Product` (D2). After some pre-manufacturing steps (S2), sub-assemblies which are available on stock are taken and assembled (M2). Finally, finished products are delivered to the consumer. In accordance with the ordinary situation it is assumed that the time of procurement is guaranteed by contracts with a certain degree of variability. Sub-assemblies received from the suppliers are stored separately until they are used for processing. Storage capacities are limited due to a given, finite amount of space.

Figures 7 and 8 show the corresponding *Process Chain* model. As shown in figure 7, the top level FE contains two sources and thus it appears that two chains are used to model retail customers and web customers. It is obvious, that both chains appear similar except for the first PCEs modeling the customer contact. At first each chain forks into two subchains with a probabilistic selection to decide whether a customer inquiry yields an order or not. In the latter case the chain requires no further consideration and reaches a sink-node. The web customer channel does not provide technical

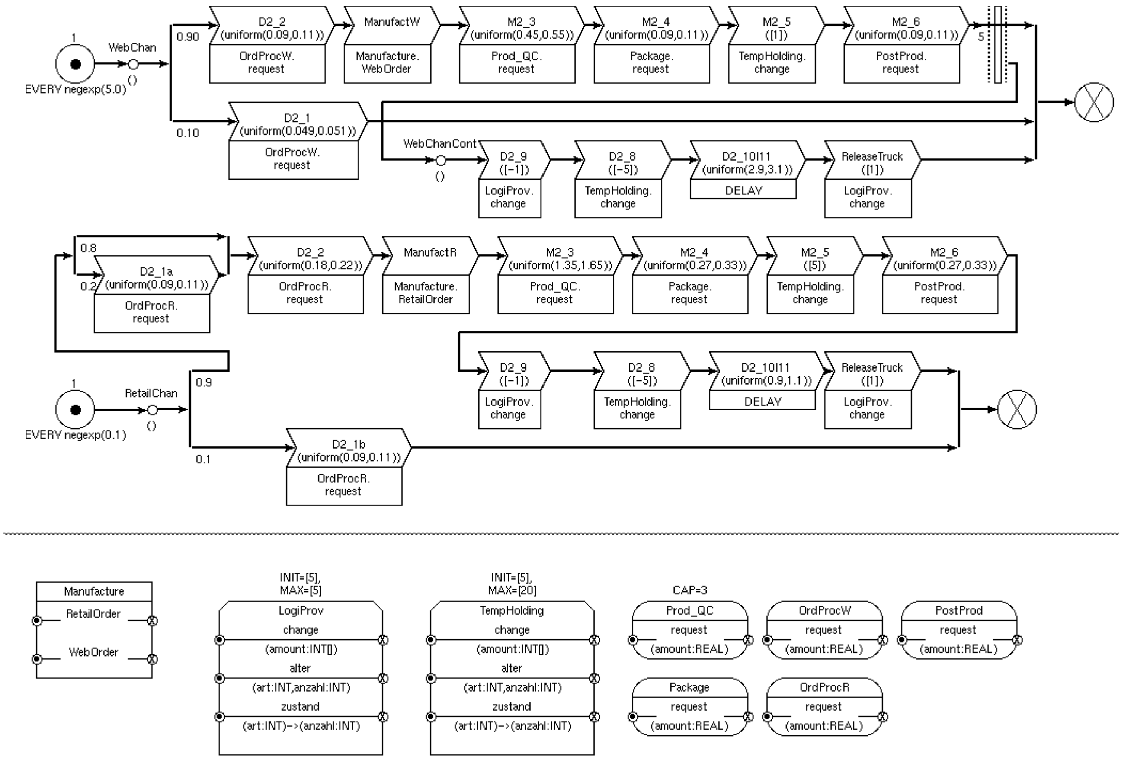

Figure 7: Top-Level FE including 2 SCs

advice or room to negotiate, only an automatic order processing system supporting a predefined set of customer information inquiries (`D2_1`) is installed. In contrast, the retail channel necessitates a further refinement of customer types whose inquiries yielding an order, because either of them claims for technical advice or negotiation, see PCE (`D2_1a`).

In any case, inquiries which result in an order are more interesting and contain the different activities performed by the SC. The depiction of underlying activities is encoded according to the SCOR terminology, for example all activities concerning the receive of orders are coded by `D2_2`. The remaining SCOR notations for activities are given in table 2.

Beside process chain patterns the model contains several (subordinated) FEs of type server, counter, store and one user-defined FE `Manufacturer`. Execution of activities usually enforces the use resp. consumption of resources and PCEs specify in detail how resources are used. For example figure 2 depicts the labeling of a particular PCE: activity `D2_1` is performed by a resource labeled as `OrderProc`. In this hypothetical case the processing of activity `D2_1` takes an amount of time described by a random variable with negative exponential distribution and mean value of 1/90 units of time. However, a process which contains this element may experience a delay which is prolonged by additional waiting time due to congestion on a possibly shared resource `OrderProc`.

In the following we do not describe each PCE in detail and highlight only important observations: 1) Both chains interact due to shared resources. 2) Most of resources are shared, only `OrdProcW` and `OrdProcR` are made available for exclusive use. All resources are listed in table 3. 3) The `WebChan` process chain comprises an order combination connector which is depicted as vertical bar. We assume, that due to its size an shipping of individual web orders is too costly, and therefore five orders are combined for one shipment operated by the out-bound logistic provider. 4) PCEs `ManufacW` and `ManufacR` invoke sub-chains `WebOrder` and `RetailOrder`, see figure 8, which are hierarchically defined below in the subordinated FE `Manufacturer`. Figure 8 shows the subchain `RetailOrder` which branches after synchronization into 3 sub-chains representing the manufacture process and two procurement processes. Furthermore the figure shows - separated by a dashed line - two passive resources modeling the inventories of supplied sub-assemblies. Capacities of inventories are assumed to be restricted by the quantity 12 and the initial value is equal to 2. The second subchain `WebOrder` has the same topology and differs only slightly from the first (different parameterizations). The access to passive resources may induce blocking and synchronization effects if no resources are available.

ACTIV.	DESCRIPTION
S2_1|2|3	Receive + verify sub-assemblies
S2_4	Store sub-assemblies (inventory)
D2_1	Respond to customer inquiries
D2_2	Process order
D2_8	Pick product from stage
D2_9	Ship product to cust. (outb. log.)
D2_10|11	Receive, test and install product
M2_1	Preproc. steps
M2_2	Issue sub-assemblies from store
M2_3	Produce and test
M2_4	Package product
M2_5	Stage product
M2_6	Release product to deliver

Table 2: Predefined activities in SCOR and their encoding

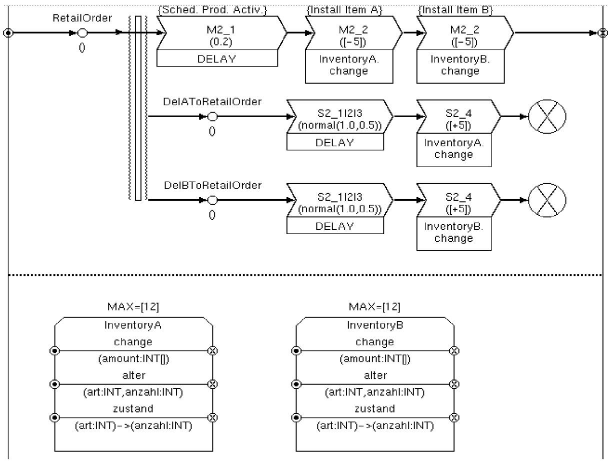

Figure 8: Subchain `RetailOrder` from the subordinated FE `Manufacturer`

RES. IDENT.	DESCRIPTION
`OrdProcW`	Web-order processing system
`OrdProcR`	Retail-order processing system
`InventA`	Store sub-assemblies A
`InventB`	Store sub-assemblies B
`Prod_QC`	Manufact. + quality control
`Package`	Packaging operation
`TempHold`	Store + stage finished product
`PostProd`	Release product to deliver
`LogiProv`	Out-bound logistic provider

Table 3: Resources and their notations

So far we have identified all activities that consume resources and therefore incur costs. The costing model necessitates to develop a quantifiable relationship between the utilization of resources and the performance of activities on one hand and to assess related costs on the other hand. In the following we discuss the running example with regard to cost origins and quantify their corresponding cost parameters.

r=`OrdProcW`: Customer inquiries (without any order) solicit quotations and request for catalogues and price-lists etc with mean costs (incurred by e-mail or postal shipping) of € 2 per request (D2_1) and hence $\mathrm{PC}_c^*(r) = 2$. Received orders are processed and maintained (D2_2) automatically, therefore the corresponding costs can be neglected.

r=`OrdProcR`: Customer inquiries and orders are received by telephone or postal mail and subsequently processed by sales people. Incurred labor costs are € 25 per hour, i.e., $\mathrm{FC}_t^*(r) = 25$. We distinguish between customer contacts resulting in orders (D2_1a) and customer contacts resulting in shipments of informative literature (D2_1b). Received orders are processed and maintained automatically (negligible costs), but the sales person receives a commission of € 3, hence $\mathrm{PC}_c^*(r) = 3$. The shipment of informative literature causes costs of € 3, so that holds $\mathrm{PC}_c^*(r) = 3$.

r=`InventA`: Each storage operation (S2_4) as well as each release operation (M2_2) costs € 1 and hence $\mathrm{PC}_c^*(r) = 1$.

r=`InventB`: To simplify matters we assume that all formulas given for `InventA` are likewise valid for `InventB`.

`WebOrder` and `RetailOrder` (processes): The supplier claims full payment of delivered subassemblies. Delivery prices are contractually guaranteed. We assume a basic charge of € 9 plus € 20 for one subassembly of type A resp. B and. Costs are assigned to their supply chains processes `WebOrder` and `RetailOrder`

r=`Prod_QC`: Machinery engaged for production and test causes fixed costs for leasing, i.e., $\mathrm{FC}_t^*(r) = 20$. Costs for the consumption of electricity and asset depreciation are assumed to be dependent of the mean processing time $t(r, M2\_3)$ and given by the parameter $\mathrm{PC}_t^*(r) = 25$.

r=`Package`: Costs for the consumption of electricity and asset depreciation are assumed to be dependent of the mean processing time $t(r, M2\_4)$ and given by the parameter $\mathrm{PC}_t^*(r) = 15$.

r=`TempHold`: Quick access to stored products requires to re-sort the inventory hourly. Expenses incurred by re-sort operations depend on the the mean number of stored products $\rho(r)$ and the costs of € 2 per stored product and hence we obtain $\mathrm{OC}_t^*(r) = 2$. Each storage operation (M2_5) as well as each release operation (D2_8) costs € 1 and hence we obtain $\mathrm{PC}_c^*(r) = 1$.

r=`PostProd`: Activities associated with post-production documentation, certification etc. cost € 3 per hour, additionally employee carry an activity-related wage of € 5 per requested working hour. Hence we obtain $\mathrm{PC}_t^*(r) = 8$.

r=`LogiProv`: Overall fixed costs for truck insurances and drivers personnel costs are equal to € 25 per hour, i.e., $\mathrm{FC}_t^*(r) = 25$. The estimated consumption of gas per truck and hour is equal to € 10. Therefore operational costs for trucks are given by the parameter setting $\mathrm{OC}_t^*(r) = 10$. In addition we charge time dependent performance costs of $\mathrm{PC}_t^*(r) = 30$ per requested working hour for several cost drivers which should not specified here.

In the remaining part of this section results of performance and costs analysis are represented. It should be noticed, that our costing approach provides a huge account of costs, because the model consists of 9 functional entities (resources) and 29 process chain elements (activities) using a resource. For each resource we obtain operational, fixed and performance costs. Furthermore performance costs with regard to all activities are at hand. Due to space limitations we restrict ourselves and present only important aggregated cost measures derived from detailed results and we encourage the reader to visit a detailed representation of cost measures available via the WWW[3]. Table 4 shows values of selected performance measures. Each column corresponds to a certain scenario as given in table 1. Surprisingly, the mean delivery time T_W for web-orders decreases although the rate (load) of web orders increases, see first row. Each shipment requires exact 5 finished products and therefore the delivery of finished products is delayed until the

PERF. ATTRIBUTE	1	2	3
T_W	14.3	8.9	7.4
T_R	4.3	4.5	4.68
ρ(`OrdProcW`)	4.7	11.9	19.0
ρ(`OrdProcR`)	20.3	17.7	14.6
ρ(`InventA`)	9.0	9.0	9.0
ρ(`InventB`)	9.3	9.3	9.3
ρ(`Prod_QC`)	52.8	57.0	61.5
ρ(`Package`)	31.7	34.3	36.9
ρ(`TempHold`)	33.1	33.9	34.7
ρ(`PostProd`)	31.7	34.3	36.9
ρ(`LogiProv`)	23.5	28.9	34.2

Table 4: Technical performance results: $T_W[T_R]$= mean time of deliveries for Web-orders [Retail-orders], $\rho(\mathtt{r})$ = utilization of resource `r`

---

[3] url: `http://ls4-www.informatik.uni-dortmund.de/home/fischer/SCM/main.html`

freight hold is filled up. All resource utilizations range between 4.7 percent and 61.5 percent. With the exception of resource `OrdProcR`, the drift of `Retail-Consumers` towards `Web-Consumers` (Scen 1 to 3) affects a higher utilization of resources, because a smaller number of larger orders is processed more efficiently compared to a larger number of smaller orders.

Table 5 shows values of selected aggregated cost measures. Increased performance costs (scenario 2 resp. 3) mainly result from increased utilizations resp. lead times. Similarly, increased operational costs are mainly affected by an increased utilization of the out-bound logistic provider `LogiProv`. As expected, fixed costs remain unchanged, because they are performance-independent. The performance costs of orders are € 44.8 resp. € 190.6 and remain also unchanged.

Cost Attribute	1	2	3
PC_t	190.0	194.1	198.6
OC_t	51.5	53.9	56.7
FC_t	70.0	70.0	70.0
PC_c(`WebChan`)	44.8	44.8	44.8
PC_c(`RetailChan`)	190.6	190.6	190.6

Table 5: PC_t=overall perf. costs, OC_t= overall operational costs, FC_t= overall fixed costs, PC_c(`WebChan`) perf. costs per Web-order, PC_c(`RetailChan`) perf. costs per Retail-order

5 Conclusion

The employed framework has been emerged as suitable and practicable approach for the modeling, analysis and costing of SCs. Current and ongoing work is dedicated to integrate in parallel existing software implementations for modeling (graphical user interface), analysis and costing into a user-friendly software package.

References

[1] A. Arantes, C. Fernandes, A.P. Guedes, and I. Themido. Logistic costs case study - an activity costing based approach. *J. of the Operational Research Society*, 51:1148–1157, 2000.

[2] M.W. Barnett and Ch.J. Miller. Analysis of the virtual enterprise using distributed supply shain modeling and simulation: an application of e-SCOR. In *Proc. of the 2000 Winter Simulation Conf.*

[3] H. Bartsch and P. Birkenbach. *Supply Chain Management with SAP APO (in German only).* Galileo Press, 2001.

[4] H. Beilner, J. Mäter, and C. Wysocki. The Hierarchical Evaluation Tool HIT. Short Papers and Tool Descript. of 7th Int. Conf. on Modelling Techniques and Tools for Computer Perf. Evaluation, Vienna (Austria), 1994.

[5] G. Bolch, S. Greiner, H. Meer de, and K. Trivedi. *Queueing networks and Markov chains.* J. Wiley, 1998.

[6] C. Cassandras. *Discrete event systems: modeling and performance analysis.* Irwin, Aksen, 1993.

[7] H. Beilner F. Bause. Beispiel zur kanonischen Berechnung von Kosten in B2. *Interner SFB-Bericht, Teilprojekt M1*, 2000.

[8] M. Voelker F. Bause, H. Beilner. A framework for the modelling and simulation of logistic networks. *4th International EUROSIM Congress, Delft, Netherlands*, 2001.

[9] A. Gunasekaren, D.K. Macbeth, and R. Lamming. Modelling and analysis of supply chain management systems: an editorial overview. *J. of the Operational Research Society*, 51:1112–1115, 2000.

[10] Th. Köpp. Visualization of performance and cost measures for logistic systems derived from process chain paradigms. Master's thesis, Fachbereich Informatik, University of Dortmund, 2001.

[11] A. Kuhn. *Prozessketten in der Logistik, Entwicklungstrends und Umsetzungsstrategien (in German).* Verlag Praxiswissen, Dortmund, 1995.

[12] D.M. Lambert, M.C. Cooper, and J.D. Pagh. Supply Chain Management: Implementation Issues and Research Oppotunities. *Int. Journal of Logistic Management*, 9(2), 1998.

[13] P. Kemper M. Arns, M. Fischer and C. Tepper. Supply Chain Modelling and its Analytical Evaluation. *accepted for or the OR 43 Conference of the Operational Research Society (will take place in Bath (UK)*, 2001.

[14] SCOR. *Supply-Chain Operations Reference-Model, SCOR version 4.0.* Supply-Chain Council, 2000.