Modeling and Evaluation of Supply Chains with GSPN Components

Gabriel Alves Jr.
Centro de Informatica -
Universidade Federal de
Pernambuco (UFPE)
Recife-PE, Brazil
gaaj@cin.ufpe.br

Paulo Maciel
Centro de Informatica -
Universidade Federal de
Pernambuco (UFPE)
Recife-PE, Brazil
prmm@cin.ufpe.br

Ricardo Lima
Departamento de Sistemas
Computacionais -
Universidade de Pernambuco
(UPE)
Recife-PE, Brazil
ricardo@dsc.upe.br

ABSTRACT

The global competition between manufactures and the requisition of high service levels for customers, led companies to recognize the relevance of supply chain management. The outbound logistics deals with the transportation and storage of finished goods from the end of the manufacturing to the customers. This paper proposes the use of pre-defined Petri nets components to model and evaluate this activity. Aiming supporting a modeling process based on library's components, a modeling tool was implemented. It allows one representing supply chains by considering a high-level model, which is further converted into a *Generalized Stochastic Petri Nets* (GSPN) model, regarding libraryŠs components. This tool was also integrated with TimeNET, a well known Petri net tool, aiming computing model's metrics. Afterwards, it is presented a real case study conducted in a Brazilian meat processing industry.

Keywords

Logistics, Modeling, Performance evaluation, Petri nets, Stochastic processes, Supply Chain Management

1. INTRODUCTION

The global competition between manufactures and the requisition of high service levels for customers, led companies to recognize the relevance of supply chain management [20]. Optimizing the supply chain can conduct a reduction of costs and improve the Quality of Service (QoS).

The outbound logistic, also referred as logistical distribution, is one of the most complex and expensive activity in the supply chain management. It concerns about the different operations required to distribute finished goods from the manufacture to customers. This activity requires decisions in the use of distribution channels, transportation modal, distribution strategy and so on. The supply chain can have warehouses, wholesalers and others, which are considered for distributing goods to customers. When analyzing the supply chain it is necessary to consider these entities.

The inventory management is also directly related to the distribution. The replenishment strategy of intermediary entities should impact the overall supply chain QoS. A well known problem that can occur in a supply chain is the *Bullwhip Effect* [11]. It results from the increasing of orders fluctuation when they move across the supply chain. In the sixties, the Sloan School of Management, Massachusetts Institute of Technology (MIT), created the *Beer Game* [20] to illustrate the Bullwhip Effect along the supply chain. In [14], this game was modeled via a timed hierarchical colored Petri Net. The work also makes experiments to evaluate the impact of different replenishment strategies over individual participants and the whole supply chain.

A new kind of Petri nets, called Batch Deterministic Stochastic Petri Nets (BDSPN), is proposed in [4] to model and evaluate the supply chain. It extends DSPNs [16] with the new concept of batch places and tokens. This new kind of places and tokens are used to model information flows while financial and material flows are represented by standard places and tokens. The batch places have an ordered set of non-negative integers for the batch tokens. Each element of the set represents a token with the size of the non-negative integer. Batch places can be used, for example, to represent arrival of customer orders. Since it is a new proposal, few works were done using this kind of net. Furthermore, there is a lack of tools that supports BDSPNs.

Generalized Stochastic Petri Nets (GSPN) [15] have been widely used for performance evaluation. This work proposes a bottom-up approach to model outbound logistics networks based on GSPN components. The supply chain scenario is modeled by the composition of these pre-defined models. Some metrics, which will be presented in latter sections, can be obtained from these models to make the evaluation of the logistical distribution [3].

The remainder of this paper is organized as follows. Section 2 gives a brief introduction to the Petri nets formalism, focusing on GSPN nets. Section 3 presents the proposed GSPN components created to evaluate the outbound logistics. The composition between these components is depicted in Section 4. A high-level modeling tool is presented in Section 5. Case studies have been conducted and one of them is presented in Section 6. Concluding remarks are given in Section 7.

Valuetools '07, October 23-25, 2007, Nantes, France

2. INTRODUCTION TO PETRI NETS

Time is an important aspect to be modeled in logistic distribution problems. There are several Petri net extensions that represent time. Some extensions associates a deterministic delay to transitions. Others associates time intervals to transitions. There are still extensions that bounds exponentially distributed delays to transitions. The Generalized Stochastic Petri Net (GSPN) [1, 15] belongs to this group. This work adopts the GSPN definition depicted in [1].

Definition 1. A Generalized Stochastic Petri Net (GSPN) is a 7-tuple defined as $GSPN = \{P, T, \pi, I, O, H, \mu_0, W\}$, where:

- P is the set of places;
- T is the set of immediate and timed transitions, $P \cap T = \emptyset$;
- $\pi : T \to \mathbb{N}$ is the priority function, where:
  $$\pi(t) = \begin{cases} \geq 1, & \text{if } t \text{ is an immediate transition;} \\ 0, & \text{otherwise.} \end{cases}$$
- $I, O, H : T \to Bag(P)$ are the input, output and inhibition functions, respectively;
- $\mu_0 : P \to \mathbb{N}$ is the initial marking function;
- $W : T \to \mathbb{R}$ is the weight function, that represents the weight (w_t) of immediate transitions and the rate (λ_t) of timed transitions, where:
  $$W(t) = \begin{cases} w_t \geq 0, & \text{if } t \text{ is an immediate transition;} \\ \lambda_t > 0, & \text{otherwise.} \end{cases}$$

As depicted in Definition 1, I, O and H are functions that given a transition $t \in T$, they retrieve a multiset of P ($Bag(P)$), where $Bag(P) : P \to \mathbb{N}$. Furthermore, it is syntactically correct to adopt $I(t, p)$ to retrieve the multiplicity of p in the multiset $I(t)$. This syntax can be used for I, O and H functions.

In a GSPN, when two or more conflicting timed transitions are enabled at the same time, this conflict is solved through a *race policy* [15]. If immediate transitions are in conflict, the choice of which to fire takes into account the respective *priority* and *weight* levels. Suppose two immediate transitions $t1$ and $t2$ in conflict. If the *priority* of $t1$ is greater than $t2$, it will always fire first. If they have the same priority, the conflict is probabilistically solved considering transitions *weights*. For a detailed description of GSPN model, the reader is referred to [15].

Definition 2. The set of input, output and inhibition places of a transition $t \in T$ are respectively ${}^\bullet t = \{p \in P | I(t, p) > 0\}$, $t^\bullet = \{p \in P | O(t, p) > 0\}$ and $t^\circ = \{p \in P | H(t, p) > 0\}$.

Definition 3. The set of output, input and inhibited transitions of a place $p \in P$ are respectively $p^\bullet = \{t \in T | I(t, p) > 0\}$, ${}^\bullet p = \{t \in T | O(t, p) > 0\}$ and ${}^\circ p = \{t \in T | H(t, p) > 0\}$.

Definition 4. A place $p\_ \in P$ is the dual place of $p \in P$ iff: $(p\_^\bullet = {}^\bullet p) \wedge ({}^\bullet p\_ = p^\bullet) \wedge (O(t, p\_) = I(t, p)) \wedge (I(t, p\_) = O(t, p)), \forall t \in T$.

A common technique known as *phase approximation* [5] can be applied to GSPNs to represent expolynomial distribution functions such as Erlang, Hipo-Exponential and Hiper-Exponential distributions, for example.

3. MODELS AND METRICS

This work proposes the use of Petri nets with stochastic times to model outbound logistics networks. One important concern is keeping the model as simple as possible in order to tackle the state-space size and simulation time. This work does not deal with routing problems; its focus is on logistical network configuration and resources utilization, besides considering QoS requirements.

In this work, the involved entities are modeled as components, allowing translating a high-level representation directly into a Petri net model. Therefore allowing hereafter modeling complex problems, without a deep knowledge of Petri nets.

This work considers the utilization of a vertical distribution channel. In this kind of channel, the information and products flows occurs between the just near entities in the channel. Every timed transition of a model, must have a delay expressed in the same time unit. One should note that for obtaining a better visualization, places, transitions, markings and arcs weights of figures are depicted without the indices used in the formal definitions.

3.1 Basic Models

We will discuss the models for the three major entities (factories, customers and warehouses) involved in the supply chain, as well as the information and goods flows. Some auxiliary models are also presented.

3.1.1 Factory

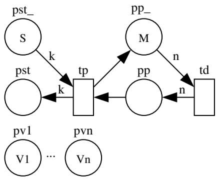


Figure 1: GSPN model for a factory.

Each factory model (see Figure 1) is a GSPN defined as $FC_i = \{P^{FC_i}, T^{FC_i}, \pi^{FC_i}, I^{FC_i}, O^{FC_i}, H^{FC_i}, \mu_0^{FC_i}, W^{FC_i}\}$, where $i = 1, 2, ..., j$. Place pst^{FC_i} represents the factory inventory and $pst\_^{FC_i}$ is its dual place. The initial marking of place $pst\_^{FC_i}$ depicts the factory's maximal storage capacity. The place pp^{FC_i} depicts producing orders, where each order should replenish the inventory with k products. Place $pp\_^{FC_i}$ is the dual place of pp^{FC_i}.

For each kind of vehicle, there is a place pvx^{FC_i}. Its initial marking denotes the available quantity of the respective vehicle category. When the orders are shipped to a consumer, the quantity of tons removed from store, must be at most equals to the maximum load capacity of the used vehicle.

The transition td^{FC_i} firing represents an internal order for producing n goods. Transition tp^{FC_i} represents the producing of goods. The rate of this transition must respect the maximal producing capacity of the factory. For instance, if a token in pst^{FC_i} represents a ton and the factory has a maximal producing rate of 2 tons per unit of time, then $\lambda_{tp^{FC_i}} = k/2$, where k is the replenishment amount whenever tp^{FC_i} fires.

3.1.2 Zone

In a supply chain, there are often dozens and even hundreds of customers. When analyzing the outbound logistics, it is reasonable to group clients into zones. A zone is generally composed of several clients of the same geographical region, which have similar consumption characteristics. It is also usually served by the same kind of vehicles with a specified periodicity. Each zone model (see Figure 2) is a GSPN defined as $ZN_i = \{P^{ZN_i}, T^{ZN_i}, \pi^{ZN_i}, I^{ZN_i}, O^{ZN_i}, H^{ZN_i}, \mu_0^{ZN_i}, W^{ZN_i}\}$, where $i = 1, 2, ..., j$.

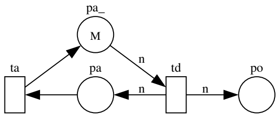


Figure 2: GSPN model for a zone.

The place po^{ZN_i} represents a recent order of the zone. Place pa^{ZN_i} represents the orders that have not yet been delivered to the zone and $pa\_^{ZN_i}$ is its dual place. $pa\_^{ZN_i}$ also ensures that the final model is bounded, thus generating a finite state-space. If its marking reaches zero in any reachable state, the zone's demand should be inhibited, what is not desired. Therefore, its initial marking (M^{ZN_i}) must be high enough for avoiding this situation.

Since firing of transition td^{ZN_i} removes n tokens from place $pa\_^{ZN_i}$, and its marking should never reach zero, a higher throughput of this transition requires a higher M^{ZN_i}. Another constraint is that $M^{ZN_i} \geq n$. If this constraint is not taken into account, transition *td* will never fire. The transitions td^{ZN_i} and ta^{ZN_i} firings represent the zone's demand and an order arrival, respectively.

In many practical situations, it is common a zone sending small orders to a producer, which accumulate them until reaching a predeterminate amount of c tons or items. The value of c is often a quantity near to the complete load of the kind of vehicle allocated to the zone.

Although possible modeling arrival of small orders, this work considers arrival of *batch orders*, each of which amounting a c quantity. This approach reduces the state-space size without loss of expressiveness, for this particular context. The occurrence of transition td^{ZN_i} denotes the zone's periodicity to form n batch orders of size c. Generally, n is equal to 1, but it is possible that a zone requires two complete loads of c units at the same time.

3.1.3 Warehouse

Warehouse model can be used not only to represent warehouses of the supply chain. This model can also represent distributors, wholesalers and so on. For instance, it is possible to have a warehouse model connected to another one, representing the supplying relationship between a distributor and a wholesaler.

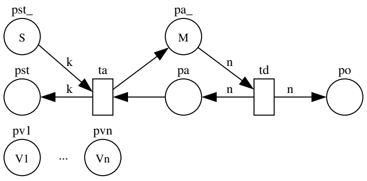


Figure 3: GSPN model for a warehouse.

Each warehouse model (see Figure 3) is a GSPN defined as $WR_i = \{P^{WR_i}, T^{WR_i}, \pi^{WR_i}, I^{WR_i}, O^{WR_i}, H^{WR_i}, \mu_0^{WR_i}, W^{WR_i}\}$, where $i = 1, 2, ..., j$.

The warehouse has characteristics of a zone and a factory. It acts like zones to the entities that supply its demands, and like a factory to entities, it provides products, but it never produces goods as a factory. Every explanation made to zone and factory models are valid to warehouse model. The warehouse transition ta^{WR_i} has an equivalent meaning of factory transition tp^{FC_i} and zone transition ta^{ZN_i}. In warehouses, the occurrence of transition ta^{WR_i} represents arrival of k items for replenishment. Furthermore, k must be equal to the shipped load per travel to the warehouse. This amount is represented in the flow component.

3.1.4 Flow

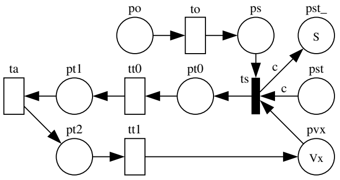


Figure 4: GSPN model for the information/goods flow.

The flow model represents the information flow from a customer to a supplier and the products flow from a supplier to a customer. This model (see Figure 4) is a GSPN defined as $FL_i = \{P^{FL_i}, T^{FL_i}, \pi^{FL_i}, I^{FL_i}, O^{FL_i}, H^{FL_i}, \mu_0^{FL_i}, W^{FL_i}\}$, where $i = 1, 2, ..., j$. Places pst^{FL_i}, $pst\_^{FL_i}$ and pvx^{FL_i} have the same meaning of producers homonymous places. When composing models, these places will be merged with their respective places of a producer. Furthermore, place pvx^{FL_i} depicts the kind of vehicle considered for serving the consumer. Place po^{FL_i} has the same meaning of customer homonymous place and will also be merged with its respective place of customer. These compositions are depicted in Section 4. A token in places $pt0^{FL_i}$, $pt1^{FL_i}$ and $pt2^{FL_i}$ indicates that a vehicle is either on the route to the consumer, delivering goods throughout, or on the route back to the producer. Place ps^{FL_i} depicts the orders that have not been shipped to the consumer yet, due to the lack of vehicles or inventory (backorders).

The transition ts^{FL_i} firing models the shipping products to a consumer. When it fires, c tokens are consumed from place pst^{FL_i}, meaning the removal of c items from producer's store. It cannot be higher than the maximal load capacity of the kind of vehicle used to send products to the consumer. Immediate transition ts^{FL_i} allows modeling of *priority* ($\pi^{FL_i}(ts^{FL_i})$) and *weight* ($w_{ts^{FL_i}}$) between consumers. The use of *priority* implies that if a zone A has higher priority than zone B, when they are in conflict the products will always be shipped to zone A. It is a useful modeling feature, since, in practice, producers usually prioritize some consumer classes. If A and B are the only conflicting zones with the same priority, the conflict is probabilistically solved considering their respective *weights*.

The occurrence of transitions to^{FL_i}, $tt0^{FL_i}$, ta^{FL_i} and $tt1^{FL_i}$ models the actions of receiving an order from a customer, traveling from producer to consumer, delivering of goods to consumer and traveling back to producer, respectively. In a real situation, it is possible making more than one order to the producer, or have more than one vehicle traveling from/to a consumer at the same time. Therefore, the depicted transitions must consider the *infinite-server semantics* (ISS) [15]. Considering ISS, transition rates increase so as the respective transition enabling degree. The clock variable related to each transition enabling runs down in parallel until reaching zero.

3.1.5 Startup Delay

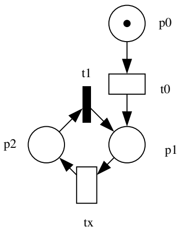


Figure 5: GSPN model for a startup delay.

Each startup delay model (see Figure 5) is a GSPN defined as $SD_i = \{P^{SD_i}, T^{SD_i}, \pi^{SD_i}, I^{SD_i}, O^{SD_i}, H^{SD_i}, \mu_0^{SD_i}, W^{SD_i}\}$, where $i = 1, 2, ..., j$. It can be considered for delaying the first firing of a transition. It is specially useful, when considering transient evaluation of the Petri net model. The enabling of transition tx^{SD_i} should only occur after the firing of transition $t0^{SD_i}$. For instance, it is possible defining that a zone should only start asking for goods after a given delay, by composing this model with a zone model and merging their respective tx and td transitions.

3.1.6 Timer

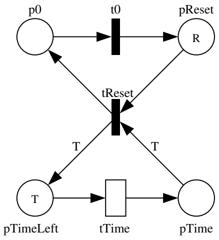


Figure 6: GSPN model for a timer.

The timer model (see Figure 6) is a GSPN defined as $TM = \{P^{TM}, T^{TM}, \pi^{TM}, I^{TM}, O^{TM}, H^{TM}, \mu_0^{TM}, W^{TM}\}$. It represents a timer with T time units, where $\mu_0(pTimeLeft^{TM}) = T$. The initial marking of place $pReset^{TM}$ can be considered for making the timer cyclic or acyclic. The timer is cyclic if $\mu_0(pReset^{TM}) > 0$, and acyclic otherwise. If the model is cyclic, once all tokens are removed from $pTimeLeft^{TM}$, transition $tReset$ fires, removing all the tokens from $pTime^{TM}$ and inserting them into $pTimeLeft^{TM}$. Thus, resetting the timer to its initial state.

This model should be considered either for modeling customer's demands, shipment enabling of orders in specific instants of time, or to control the production of goods in the factory. This model should be carefully used, since the related state-space size might considerably increase, specially when $pTimeLeft^{TM}$ has a high number of tokens at its initial marking as well as considering a cyclic timer

The immediate transition $tReset^{TM}$ should have the lowest priority of the resultant model. Suppose an immediate transition of a flow model that considers the markings of place $pTime^{TM}$ as a pre-condition. Whenever both transitions are simultaneously enabled, such a transition must always fire first than $tReset^{TM}$. Thus, the priority of such a transition must be higher than $tReset^{TM}$'s priority.

3.2 Metrics

Let $E\{< \ marking\_function \ >\}$ be the expected value for a marking dependent expression, and $P\{< logic\_condition >\}$ be the probability for a logic condition occurs. Where a marking function has the form $\sharp p$ and refer to the number of tokens in place p. The logic condition compares a marking function with another marking function or with an integer.

Some performance indices that can be extracted from models are depicted in Table 1. Entities columns refer to the entity from which the metrics can be extracted. The entities can be either factories (**FC**), warehouses (**WR**), zones (**ZN**), flows (**FL**) and also the composition between a flow model and a customer zone (**FLZN**) or a customer warehouse (**FLWR**). Models compositions are depicted in Section 4.

3.3 Extending Basic Models

Two additional features can be considered for extending the behavior of proposed models: marking dependent arcs weights and enabling functions for immediate transitions. The former allows expressing the arcs weights with expressions based on places markings. This feature should be carefully used since it can affect the invariants of the resultant model.

The second allows including a guard based on places markings for immediate transitions. Thus, if this guard evaluates to false, the transition gets disabled. For instance, this feature can be considered whenever the use of arcs for enabling a transition should result in a model of difficult understanding. One should bear in mind that it can lead to deadlock that cannot be structurally captured. For instance, it can occur if the expression is always evaluated to false in every reachable marking, thus the transition would never be enabled.

Several replenishment policies could be modeled considering these modeling features. For instance, considering a factory model like depicted in Figure 1, in which $k = 1$ and $n = 20 - \sharp pst^{FC_i}$ are the arcs' weights depicted in this model, and let p be the delay assigned to transition td. Due to the marking dependent function of the arc weight n, at each period of time p, an order will be created for replenishing the inventory at the level of 20 units.

Still considering the factory model, suppose that transition tp is replaced by an immediate transition with the enabling function $(\sharp pst + \sharp pp) < 15$ and $k = 10$. Once the sum of the inventory and the amount being produced gets lower then 15 units, an order will be created for replenishing the inventory with more 10 units.

The enabling function and the timer model should also be considered for disabling immediate transitions of the model. Suppose a cyclic timer with $\mu_0(pTimeLeft) = 24$ and an enabling function $(\sharp pTime > 8) and (\sharp pTime < 18)$ asso-

Table 1: Some important metrics.

Entity	Performance Indice	Metric
FC or WR	Expected Inventory Size	$E\{\sharp pst\}$
FC or WR	Empty Inventory Probability	$P\{\sharp pst = 0\}$
FC or WR	Full Inventory Probability	$P\{\sharp pst_{-} = 0\}$
FC or WR	Available Vehicles of kind pvx	$E\{\sharp pvx\}$
FC or WR	Non-Use Vehicles of kind pvx Probability	$P\{\sharp pvx = Vx\}$
FC or WR	Non-Availability of Vehicles of kind pvx Prob.	$P\{\sharp pvx = 0\}$
ZN or WR	Recent order	$E\{\sharp po\}$
ZN or WR	Pending Finishing Delivering Orders	$E\{\sharp pa\}$
FL	Pending Shipping Orders (Backorder)	$E\{\sharp ps\}$
FL	Pending Shipping Orders $> n$ Probability	$P\{\sharp ps > n\}$
FL	N^o of Vehicles Delivering to Consumer	$E\{\sharp pt1\}$
FL	In transit inventory	$E\{c \times (\sharp pt0 + \sharp pt1)\}$
FLZN	Delivering to Zone Throughput	$P\{\sharp pt1 > 0 \text{ AND } \sharp pa > 0\} \times \lambda_{ta}$
FLWR	Delivering to Warehouse Throughput	$P\{\sharp pt1 > 0 \text{ AND } \sharp pa > 0 \text{ AND } \sharp pst_{-} > k\} \times \lambda_{ta}$

ciated with immediate transitions ts of every flow model. This enabling function should model the shipping products disabling from 18:00 evening till 6:00 morning.

The startup delay model, should also have its behavior modeled considering enabling functions and the timer model. Suppose that transition $t0$ of the startup delay is replaced by an immediate transition. Considering an acyclic timer with $\mu_0(pTimeLeft) = 2$ and an enabling function $\sharp pTime = 2$ associated with $t0$, the first firing of $t0$ should occur after 2 time unit delay.

4. COMPOSING MODELS

In an outbound logistics scenario, source nodes must be represented by factory models and end nodes by zone models. The intermediaries entities must be represented by warehouse models. If costumers of a wholesaler are not to be represented in the model, then the wholesaler must be modeled as a zone.

The possible connections between entities are depicted in Figure 7. A connection is represented by a directed arc. The direction of the arc represents the flow of goods, while the opposite direction depicts the information flow. The origin of the arc depicts the producer's kind of vehicle considered for distributing goods to the destination. The arc's arrow represents the customer. For instance, Figure 7 (a) depicts a distribution of goods from a factory to a zone using vehicles of kind $V1$.

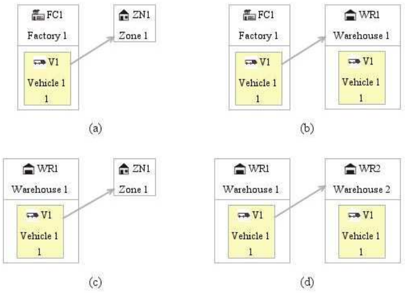


Figure 7: High-level representation of a connection from: (a) factory to a zone; (b) factory to a warehouse; (c) warehouse to a zone; (d) warehouse to other warehouse.

When translating these models into a GSPN, entities are converted into their respective GSPN component. To represent the connection, a flow component connects origin and destination components. This composition is carried through fusion of places and transitions of GSPN components.

For instance, the GSPN models for the possible connections of Figure 7 (a) and Figure 7 (b) are depicted in Figures 8 and 9, respectively. One should note that for obtaining a better visualization, arcs' weights and initial markings are depicted without the indices used in the formal definitions of Section 3. Thus, different values should be assigned to the weight k of warehouse and factory in Figure 9, for example.

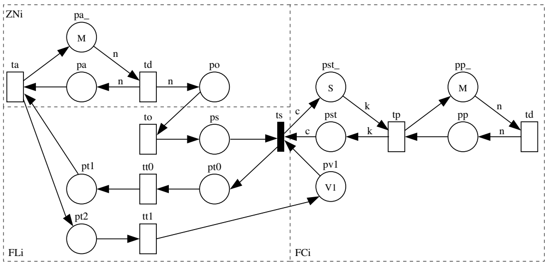


Figure 8: GSPN model for connecting a factory to a zone (Figure 7 (a)).

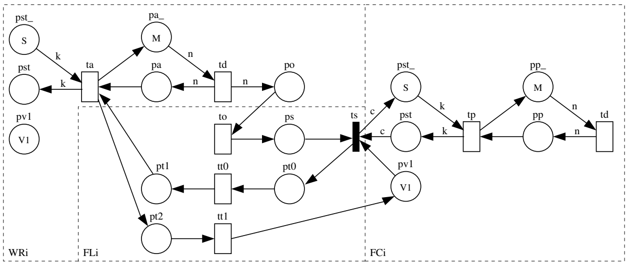


Figure 9: GSPN model for connecting a factory to a warehouse (Figure 7 (b)).

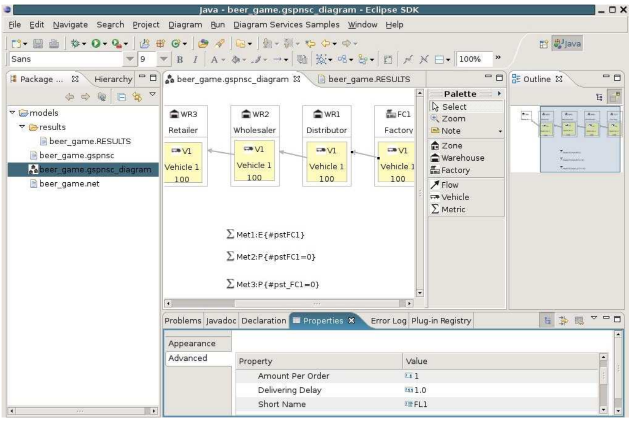
P-invariants guarantees the *structural boundedness* and *conservativeness*, thus allowing performance evaluations based on the state-space analysis. T-invariants are necessary structural conditions for the *liveness* and for *deadlock freedom* [18, 13].

The final GSPN model resulting from depicted compositions, will always be completely covered by positive P-invariants and T-invariants. Beyond the presented composition rules, some constraints must be considered. For zones and warehouses models, arc's weight n must be lower than or equals to $\mu_0(pa\_)$. For factories, it must be lower than or equals to $\mu_0(pp\_)$. For factories and warehouse models, arc's weight k must be lower than $\mu_0(pst\_)$. Considering the composition between a consumer warehouse and a flow model, arc's weight k must be equals to c. Finally, for compositions between an origin factory (or warehouse) and a flow model, c must be lower than $\mu_0(pst\_)$.

When employing a startup model for delaying the first transition firing, the model will not be completely covered by T-invariants. It occurs because this model work as a syphon [8], and once its transition *td* fires, it will never fire again. All others transitions should be covered by T-invariants.

5. SLOT - STOCHASTIC LOGISTICS OPTIMIZER TOOL

This section presents the Stochastic Logistics Optimizer Tool (SLOT), a high-level modeling tool that supports the automatic generation stochastic Petri net models for performance evaluation of outbound logistic scenarios. SLOT supports the set of models conceived for representing the diverse supply chain strategies as well the distribution policies. A library with those models has also been implemented. Furthermore, it allows the evaluation of metrics from the high-level representation. A snapshot of SLOT is presented in Figure 10.



Met1:E{#pstFC1}

Met2:P{#pstFC1=0}

Met3:P{#pst_FC1=0}

Property	Value
Amount Per Order	1
Delivering Delay	1.0
Short Name	FL1

Figure 10: SLOT Snapshot.

On the left side, a list of projects is presented. In the project folder, the respective file list is presented. In this case, two files with the same name, but with different extensions have been used. The file extension *gspnsc* is used to store the model's data, while the *gspnsc_diagram* is used to store data related to the model's visualization. At the top, there is a toolbar with buttons for actions such as font change, alignment and zoom. These and other functions can also be accessed through the *Diagram* menu.

The editor is presented in the center. In order to insert an entity or a metric into the model, the user should select the respective tool in the editor's palette and then click on the editor's area. To add a vehicle to a warehouse or a factory, the user should select the *vehicle* tool and click on the producer's *vehicles compartment* (located below the entity name). To connect entities, the user should select the connection tool, click on a producer's vehicle and then on the destination consumer.

The *Properties view*, at the bottom of the figure, shows the properties from the current selection. In the figure, the properties of the connection from the factory's vehicle *V1* to the zone *ZN2* are shown. For instance, it is possible to change the traveling time from *FC1* to *ZN2*. At the right of the figure, the *outline view* shows an overview of the editor's area. This overview facilitates navigation through large models that do not fit on the visible area.

SLOT was developed as a set of Eclipse plug-ins [6]. The Eclipse was first developed in 2001 by a consortium composed of industry leaders such as IBM, Borland and Rational. It was designed to work with plug-ins. The inclusion of new plug-ins provides it with new functionalities. An overview of Eclipse architecture is presented in Figure 11.

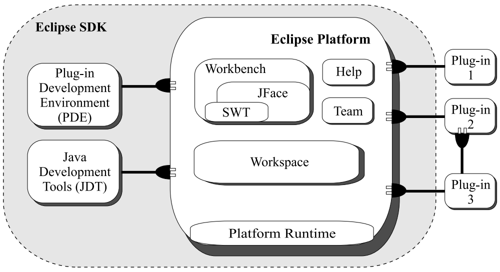


Figure 11: Overview of Eclipse architecture.

SLOT was implemented using the *Graphical Modeling Framework* (GMF) [10]. It bridges the *Graphical Editing Framework* (GEF) and the *Eclipse Modeling Framework* (EMF) [17], by creating GEF editors that display data from models managed and persisted using EMF. Based on a number of artifacts, this framework generates codification for editing and persisting models. This process is depicted in Figure 12 and was extracted from the GMF tutorial [10].

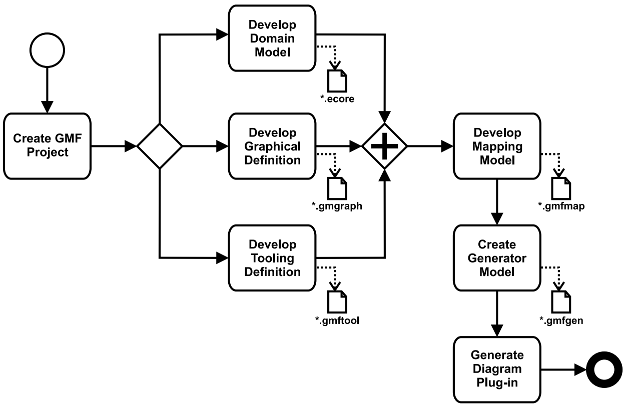


Figure 12: Generating an editor code with GMF.

The GMF considers a graphical definition model to keep information related to GEF elements that will be presented in the graphical editor. For instance, it defines figures and labels for entities and connection decorators. Vehicle compartments are also defined in it. When creating the mapping definition, these compartments are inserted into factories and warehouses figures. Thus, vehicles should be added only into these compartments. In Figure 10, the factory model has two vehicles into its compartment, while the warehouse model has one vehicle.

A tooling definition model is employed for designing the editor's palette, menus, toolbars and others. The domain model should be represented in the Ecore (EMF metamodel) format. This model was created using the *Ecore Diagram* editor, provided by GMF. This model is depicted in Figure 13. It is also the basis for generating the EMF domain model classes.

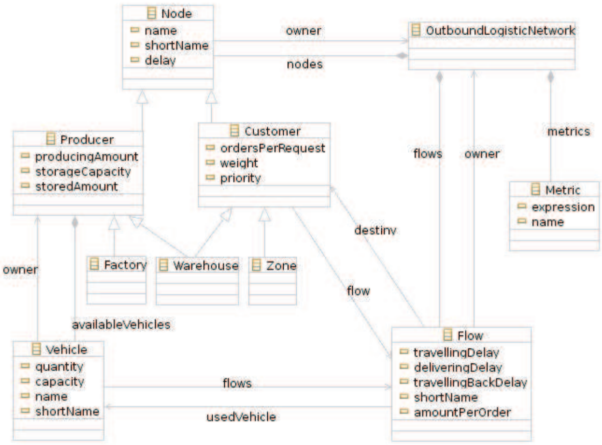


Figure 13: Ecore model for the graphical outbound logistics editor.

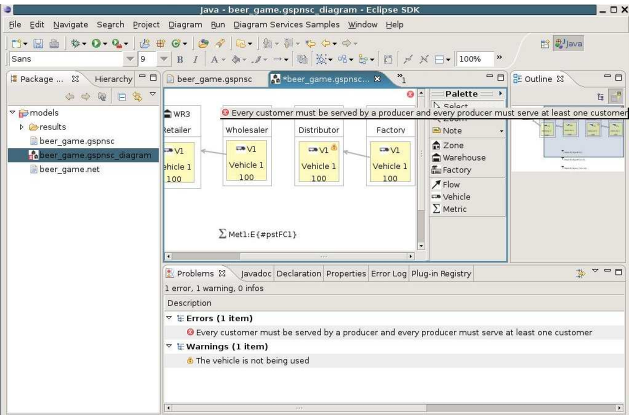
Based on these models, the GMF generates a mapping model file, bridging the domain model (EMF) to a graphical definition (GEF). Having the mapping model, GMF provides a generator model allowing implementing specific details to be defined for generating the editor plug-in. Every generated code is annotated with the *@generated* tag. When customizing the generated code, it is essential to remove the *@generated* tag or change it to *@generated NOT*. Thus, GMF and EMF should be aware to not replace customized code through generation sessions.

GMF allows the usage of *link constraints* [9] to control the creation of connections. They enable or disable the connection starting or ending based on the expressions provided. The preferred language for constructing expressions is the *Object Constraint Language* (OCL) [19]. In SLOT, connections are done from a vehicle to a customer, where the customer can be either a warehouse or a zone. The connection's target cannot be the owner of the vehicle. In the current version of SLOT, although a producer could supply many consumers, each consumer must be supplied by a single producer. To express these conditions, the following OCL expression was assigned as a *link constraint* to the target end of the connection.

```
self <> oppositeEnd.owner and self.owner.flows ->
    forAll(r | r.destination <> self)
```

Keywords *self* and *oppositeEnd* refer to the end target itself and to the connection's source, respectively. In this specific context, *self* is the destination consumer and *oppositeEnd* is the producer's vehicle used to distribute goods. Thus, expressions *oppositeEnd.owner* and *self.owner.flows* should retrieve the producer and the flows of the outbound logistic network, respectively.

GMF also supports the model validation through the use of *Audits*, which is a set of rules that must hold true when evaluated in a diagram instance. Audits severity can be *INFO*, *WARNING* or *ERROR*. Audits of SLOT were constructed using OCL expressions. The created audits aim validating entities attribute's fulfillment, to check whether there are metrics in the model or the producer's inventory is over its maximum storage capacity, for example. When running SLOT, these validations can be accessed via the *Diagram > Validate* menu. A message is presented in the *Problems view* for each audit rule evaluated to false. An icon is also presented beyond the entity to which the OCL expression refers. Figure 14 presents a snapshot after the validation of a model.



Every customer must be served by a producer and every producer must serve at least one customer

1 error, 1 warning, 0 infos

Errors (1 item)

Every customer must be served by a producer and every producer must serve at least one customer

Warnings (1 item)

The vehicle is not being used

Figure 14: Snapshot of SLOT validation.

TimeNET [7] is one of the most popular Petri tools, and has been adopted for modeling and performance evaluation of several timed Petri net classes such as GSPNs. An integration with TimeNET 3.0 was provided for opening the Petri net model for an outbound logistic model and for evaluating metrics. TimeNET 3.0 supports for Linux and SunOS operating systems. This integration was developed and tested in the Linux environment, thus it is not guaranteed to work within SunOS.

By right clicking over the editor's area, a popup menu containing the TimeNET options is exhibited. Currently, it is possible to parse the high-level model to a Petri net model, which creates an *.net* file which can be opened with the TimeNET. There is also the possibility of directly opening the TimeNET with the Petri net that represents the modeled outbound logistic network. Finally, it is possible to evaluate metrics considering stationary or transient evaluation (analysis or simulation).

To parse the modeled outbound logistic network to a Petri net format, SLOT employs the models and composing process presented in previous sections. Parameters informed in the model are also considered when assigning Petri net's elements names, delays, priorities and so on. The final Petri net model is stored in *.net* file, which can be opened with the TimeNET. This model can then be changed, applying the *phase approximation* technique, for example. Besides generating the *.net* and open it later with TimeNET, there is also an option to directly opening the TimeNET with the Petri net model.

The Java Native Interface (JNI) [12] was used for integrating TimeNET's evaluation tools into SLOT. Thus, allowing the evaluation of metrics without explicitly parsing the model or opening the TimeNET. Once evaluation is finished, these tools generate files with the metrics results. These files are copied to SLOT's active project, allowing users to visualize the metrics results.

6. CASE STUDY

This section presents a real case study conducted in a Brazilian meat processing industry (São Mateus Frigorífico). This industry has more than one thousand employees and capacity to produce more than five thousand tons of goods per month. It owns about 50 vehicles to serve more than 250 travels per month for delivering goods to clients.

The industry was planning increasing sales. Cases studies were conducted with different vehicles and zones, with the aim of investigating if the current number of vehicles was sufficient for supporting the expected rising of customers demand. This paper presents the case study taking into account zones served by 2-axle trucks. In addition to changes in the current outbound logistics network, the industry imposes a QoS of at least 98%. Thus, the backorders probability should be kept below 2%.

After an interviewing process, information about the current outbound logistics has been collected and analyzed. The direct shipping strategy [20] was adopted in this particular case study. The Pareto chart in Figure 15 depicts the amount of tons delivered to each zone with 2-axle trucks. Only zones explicitly presented in the Pareto chart were considered for analysis. That is, those in the group which consumes 95% of delivered goods. The remaining zones were discarded. The outbound logistics network was modeled with the SLOT and converted into an a GSPN model, depicted in Figure 16.

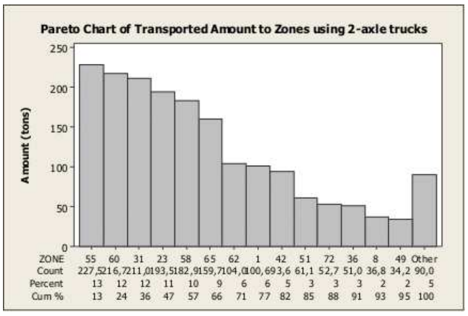


Figure 15: Pareto chart with the transported amount to zones served by 2-axle vehicles.

Afterwards, one need choosing which indices will be adopted in the evaluation. In this study, we intend to evaluate the vehicles utilization. Due to the industry QoS requirements, it is also necessary to measure the customers backorders.

Statistical analysis was performed taking into account historical data for inferring the required model parameters.

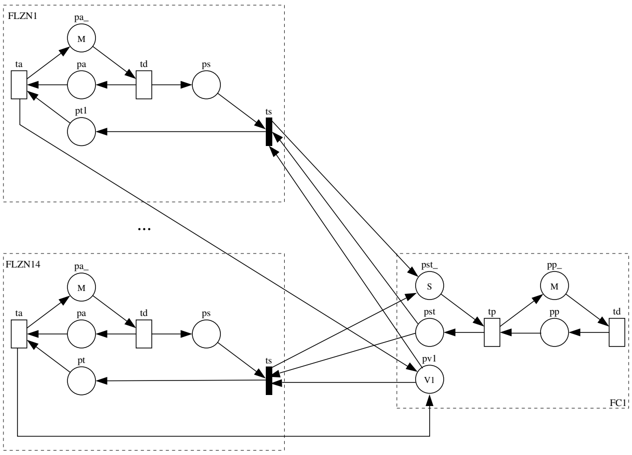


Figure 16: GSPN model for São Mateus Frigorífico outbound logistics.

Travels' duration and the interval between travels, were analyzed for each zone. Outliers of these values have been removed through the *Inter Quartile Range* (IQR) analysis. This process avoids spurious data and has important impact on the evaluation results. After that, the average and standard deviation were obtained, and employed for applying the *phase approximation* technique, and then generating the refined model.

For this case study, it is intended removing interference related with products unavailability. In order to guarantee that there will always be enough goods in the store, a high value was assigned to the production rate. Furthermore, this value respects the industry's maximum production capacity. The transported amount of goods was directly obtained from historical data. This value is assigned to an arc weight in the GSPN model. Since arcs weights cannot be decimal numbers, the transported amounts were rounded up to the nearest integer value. The resultant parameters for the analyzed zones are depicted in Table 2.

Table 2: Parameters per zone.

Zone	Short Name	N	Interval (days)		Time (hours)		c
			Mean	StDev	Mean	StDev	
55	ZN_1	26	5,62	6,84	75,65	14,94	8
60	ZN_2	24	6,458	3,107	137,17	18,07	9
31	ZN_3	24	6,458	3,176	105,58	25,02	9
23	ZN_4	26	7,27	7,43	78,38	18,99	8
58	ZN_5	22	6,091	3,069	93,73	21,95	9
65	ZN_6	22	6,955	3,848	124	29,55	7
62	ZN_7	12	11,58	6,82	82,83	18	9
1	ZN_8	16	5,38	4,11	118,44	17,05	7
42	ZN_9	11	13,27	8,84	100,45	22,58	9
51	ZN_{10}	8	13,38	6,59	90,63	13,33	8
72	ZN_{11}	11	11,38	13,09	38,3	29,5	7
36	ZN_{12}	9	8,89	6,25	37,44	17,73	6
8	ZN_{13}	4	25,8	23,3	58,8	36,4	9
49	ZN_{14}	4	15,5	14,15	94	14,02	8

In Table 2, the column **Short Name** shows the names of zone adopting the nomenclature described in previous sections. Column **N** represents the number of travels to a zone. The interval between travels and the total traveling time are shown in columns **Interval** and **Time**, respectively. The median of the transported amount per travel is presented in column **c**.

Table 3: Metrics results for scenarios when varying the customers demand.

Performance Indice	Metric	Demand Increment (%)					
		0	5	10	15	20	25
Available Vehicles	$E\{\sharp pv1^{FC_1}\}$	5.64	5.02	4.74	4.25	3.94	3.52
None Used Vehicles Prob.	$P\{\sharp pv1^{FC_1} = 12\}$	0.16%	0.12%	0.07%	0.05%	0.03%	0.02
None Available Vehicles Prob.	$P\{\sharp pv1^{FC_1} = 0\}$	3.62%	4.74%	8.23%	12.54%	13.86%	18.99
Shipping Locked by Storage Prob.	$P\{\sharp pst^{FC_1} < 108\}$	0.00%	0.00%	0.00%	0.00%	0.00%	0.00
Backorder Prob. (Zone 55)	$P\{\sharp ps^{ZN_1} > 0\}$	0.32%	0.50%	0.81%	1.25%	2.16%	3.52%
Backorder Prob. (Zone 60)	$P\{\sharp ps^{ZN_2} > 0\}$	0.28%	0.43%	0.72%	1.14%	1.85%	2.99%
Backorder Prob. (Zone 31)	$P\{\sharp ps^{ZN_3} > 0\}$	0.28%	0.46%	0.75%	1.09%	1.78%	3.01%
Backorder Prob. (Zone 23)	$P\{\sharp ps^{ZN_4} > 0\}$	0.26%	0.40%	0.63%	0.99%	1.62%	2.65%
Backorder Prob. (Zone 58)	$P\{\sharp ps^{ZN_5} > 0\}$	0.31%	0.48%	0.75%	1.20%	2.03%	3.28%
Backorder Prob. (Zone 65)	$P\{\sharp ps^{ZN_6} > 0\}$	0.27%	0.40%	0.65%	1.01%	1.75%	2.88%
Backorder Prob. (Zone 62)	$P\{\sharp ps^{ZN_7} > 0\}$	0.16%	0.25%	0.41%	0.65%	1.04%	1.77%
Backorder Prob. (Zone 1)	$P\{\sharp ps^{ZN_8} > 0\}$	0.35%	0.53%	0.85%	1.26%	2.14%	3.62%
Backorder Prob. (Zone 42)	$P\{\sharp ps^{ZN_9} > 0\}$	0.14%	0.22%	0.35%	0.59%	0.88%	1.58%
Backorder Prob. (Zone 51)	$P\{\sharp ps^{ZN_{10}} > 0\}$	0.14%	0.22%	0.34%	0.55%	0.88%	1.45%
Backorder Prob. (Zone 72)	$P\{\sharp ps^{ZN_{11}} > 0\}$	0.16%	0.25%	0.41%	0.64%	1.05%	1.79%
Backorder Prob. (Zone 36)	$P\{\sharp ps^{ZN_{12}} > 0\}$	0.21%	0.33%	0.53%	0.81%	1.29%	2.31%
Backorder Prob. (Zone 8)	$P\{\sharp ps^{ZN_{13}} > 0\}$	0.08%	0.12%	0.18%	0.29%	0.48%	0.83%
Backorder Prob. (Zone 49)	$P\{\sharp ps^{ZN_{14}} > 0\}$	0.12%	0.19%	0.30%	0.48%	0.79%	1.43%

Metrics computed from model are presented in the first column of Table 3. The first three metrics represent the expected number of available vehicles in the factory, the probability of having no available vehicles in the factory, and the probability of having none vehicle being used. The fourth metrics evaluates if the stored amount will not block the shipping of an order. The value for this metric must be close to 0. Finally, the probability of having at least one backorder is evaluated for each zone.

Although the resultant model is analyzable, the state-space might be too large to be computationally viable considering the available computer resources. The objective of this case study was to evaluate the impact of incrementing industry's sales on the presented metrics. Increments in zone's demand are modeled by decreasing the interval between travels to the zone. Different scenarios were evaluated varying the zones' demand. Results can be seen in Table 3. For a better visualization, Figure 17 depicts the results for the backorders probabilities.

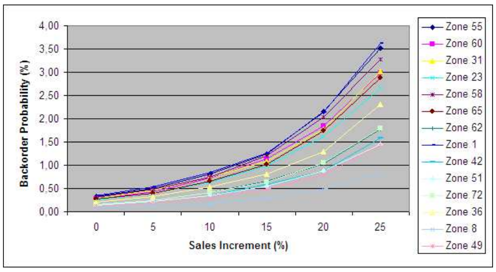


Figure 17: Backorders probability per zones varying demand.

When the zone's demand is increased in 20%, the probability for every vehicles being used rises from 3.62% to 13.86%, but the backorders probability is still close to 2%, in the worst case. Consequently, keeping the same quantity of available vehicles, this industry can increase its sales up to 20% and still keep a QoS close to 98%. Increasing sales over 20% could also be considered, by hiring a third-party distribution services in the moments of greatest demands.

7. CONCLUSIONS

This paper presented a proposal for modeling and evaluating the outbound logistics of a supply chain. A supply chain scenario can be modeled with a bottom-up approach considering the proposed GSPN models. Metrics that could be extracted from these models were presented. These metrics might be very relevant for a decision making process. For instance, it is possible to evaluate the expected number of vehicles in use, or the probability for a customer having more then x backorders.

A direct conversion from a high-level representation of an outbound logistic scenario to a GSPN model is also provided through the composition of proposed GSPN components. The use of these components, respecting composition rules presented along this paper, guarantees that the final model will have some desired Petri nets' properties by construction. Additionally, it can also be formally analyzed through well-established mathematical methods. Consequently, qualitative analysis of Petri nets' properties might be avoided, contributing to the efficiency of the evaluation process.

The Stochastic Logistics Optimizer Tool (SLOT) was implemented as a set of Eclipse plug-ins. It allows the modeling of an outbound logistic network and translate it into a GSPN model. This model can be evaluated by the TimeNET tool. An integration mechanism has been implemented to connect SLOT and TimeNET, thus allowing models' evaluation directly from SLOT.

The proposed models were applied in a Brazilian meat processing industry. The case study presented in this paper analyzes the outbound logistics network, considering zones served by a specific kind of vehicle employed in the industry. The obtained results depicted that with the current fleet, this industry could increase its sales up to 20%, keeping the QoS close to 98%.

In future studies, we intend analyzing the effects of delivery failures on the QoS. We also intend to study intermodal/multimodal transportation [2]. In such cases, management distribution considers that goods can be delivered to clients using more than one type of transportation means.

8. ACKNOWLEDGMENTS

Authors would like to thank the São Mateus Frigorífico for supporting the development of this work.

9. REFERENCES

[1] G. Balbo. Introduction to Stochastic Petri Nets. *LNCS 2090*, pages 84–pp, 2001.

[2] R. H. Ballou. *Business Logistics/Supply Chain Management*. Pearson Education, Inc., 5 edition, 2004.

[3] B. M. Beamon. Measuring supply chain performance. *International Journal of Operations & Production Management*, 19(3):275–292, March 1999.

[4] H. Chen, L. Amodeo, and F. Chu. Batch Deterministic and Stochastic Petri Nets: A Tool for Modeling and Performance Evaluation of Supply Chain. In *Proceedings of the 2002 IEEE International Conference on Robotics & Automation*, 2002.

[5] A. A. Desrochers and R. Y. Al-Jaar. *Applications of Petri Nets in Manufacturing Systems: Modeling, Control, and Performance Analysis*. Institute of Electrical & Electronics Enginee, 1994.

[6] E. Gamma and K. Beck. *Contributing to Eclipse: Principles, Patterns, and Plug-Ins*. Addison Wesley Professional, 2003.

[7] R. German, C. Kelling, A. Zimmermann, and G. Hommel. TimeNET - A toolkit for evaluating non-markovian stochastic Petri nets. Technical Report 19, Technische Universität Berlin, 1994.

[8] C. Girault and R. Valk. *Petri Nets for Systens Engineering: A Guide to Modeling, Verification and Applications*. Springer-Verlag, 1995.

[9] GMF Constraints. http://wiki.eclipse.org/index.php/GMF_Constraints. Last access on march 25, 2007.

[10] GMF Tutorial. http://wiki.eclipse.org/index.php/GMF_Tutorial. Last access on march 25, 2007.

[11] H. L. Lee, V. Padmanabhan, and S. Whang. The Bullwhip Effect in Supply Chain. *Sloan Management Review*, 38(3), 1997.

[12] S. Liang. *The Java Native Interface: Programmer's Guide and Specification*. Addison Wesley Professional, 1999.

[13] P. Maciel, R. Lins, and P. Cunha. *Introdução às Redes de Petri e Aplicações*. Campinas - SP: Sociedade Brasileira de Computação, 1996.

[14] D. Makajic-Nikolic, B. Panic, and M. Vujosevic. Bullwhip Effect and Supply Chain Modelling and Analysis using CPN Tools. In *5th Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*, 2004.

[15] M. A. Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. John Wiley and Sons, 1995.

[16] M. A. Marsan and G. Chiola. On Petri nets with deterministic and exponentially distributed firing times. In *7th European Workshop on Applications and Theory of Petri Nets*, pages 132–145. Springer-Verlag, 1987.

[17] B. Moore, D. Dean, A. Gerber, G. Wagenknecht, and P. Vanderheyden. *Eclipse Development using the Graphical Editing Framework and the Eclipse Modeling Framework*. IBM.Com/Redbooks, 2004.

[18] T. Murata. Petri Nets: Properties, Analysis and Applications. *Proc. IEEE*, 77(4):541–580, April 1989.

[19] Object Modeling Group (OMG). *Object Constraint Language, v2.0*, may 2006.

[20] D. Simchi-Levi, P. Kaminsky, and E. Simchi-Levi. *Designing and managing the supply chain: concepts, strategies and case studies*. The McGraw-Hill Companies, Inc., 2000.