A VERIFICATION METHOD FOR THE SIMULATION OF SUPPLY CHAIN NETWORKS WITH UNRELIABLE LINKS

KATRIEN RAMAEKERS and GERRIT K. JANSSENS
Hasselt University, Diepenbeek, Belgium
katrien.ramaekers@uhasselt.be, gerrit.janssens@uhasselt.be

KENNETH SÖRENSEN
Faculty of Applied Economics
University of Antwerp, Antwerp, Belgium
e-mail: kenneth.sorensen@ua.ac.be

RIK VAN LANDEGHEM
Industrial Management
University of Ghent, Ghent, Belgium
e-mail: hendrik.vanlandeghem@ugent.be

Abstract: As supply chains exceed the limits of a company their behaviour grows more complex. The design of an efficient global supply chain network is of great importance in a competitive environment. As many uncertainties occur in the flow of goods, information or payments, simulation is an appropriate tool to model the behaviour of such a network. A chain is known to be as good as its weakest link and these links have shown to be failure prone. Therefore our focus is on chains with unreliable links. This enlarges the complexity of the model. In order to obtain a correct model for the logistics reality to be studied, we propose a formal method, which is able to generate a simulation model in an automatic way. Petri nets are chosen as the formalism. The procedure to build such a model is outlined in detail for a serial logistics system.
Keywords: Verification, logistics, supply chains, Petri nets

INTRODUCTION

The supply chain of a company encompasses all entities, such as plants, distribution centres and transportation modes, that ensure a steady flow of products, from raw materials, up to the final delivery in the hands of the consumer. This means that most supply chains link multiple companies in a sequence of supplier-customer relations (Beamon 1998, Slats et al. 1995).

The literature on designing a supply chain focuses mainly on the objectives of minimising cost or maximising profit. Most methodologies do not take into account reliability, although it has been identified as a key performance contributor (Vidal and Goetschalckx 1996). A reliable supply chain should assure on-time arrival of parts and subassemblies to the different locations, so that excessive work-in-process do not builds up causing delays, inventory, costs or poor customer service. The notion of supply chain reaction time refers to the total elapsed time between receipt of the customer order and the final delivery of the complete order.

The major components of a generic supply chain are shown in Figure 1. They can de divided into: information flows (dashed line), material flows (solid line), process steps (rectangle) and storage locations (triangles). The directed network, formed by the sequence of flows in Figure 1, consists of two paths, representing the major flows through the supply chain.

A first flow (A) starts with the customer (the order), is converted by the Order Mix function (a planning step) into an assembly command that passes to the component storage. In the storage material is picked for the order at hand, and moved to the assembly line. Assembly can only start if all parts are available. Some of the parts are on-order only, which forms a second path (B): from the Order Mix, to Purchasing and the Suppliers.

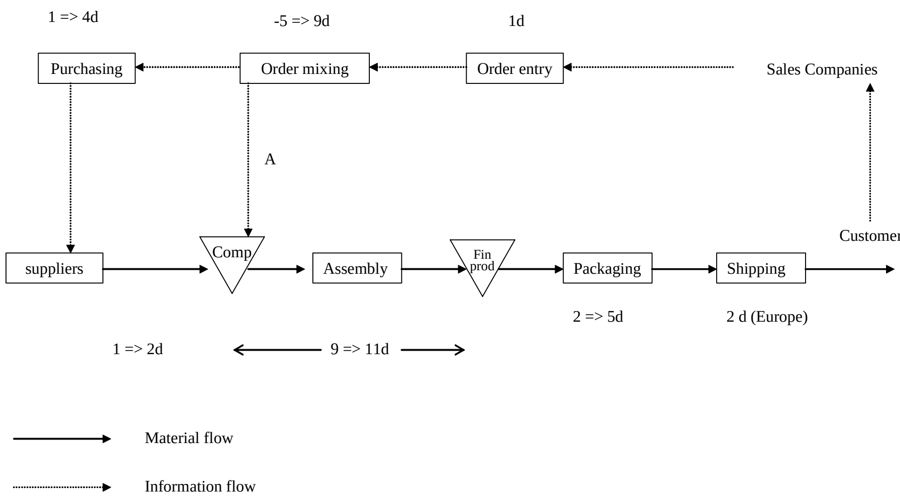

Figure 1: A generic supply chain

There the components are prepared and delivered to the assembly, allowing to finish this customer order.

This generic scenario reflects typical planning behaviour. In order to evaluate the benefits of the integration of such a chain, its performance is likely to be estimated by means of computer simulation. In practice rigourous planning is disturbed due to uncertainties and failures. Uncertainties have instantiations as variable lead times between intermediate links in the chain. They are characterised by a probability distribution. Figure 1 shows maximum and minimum lead times in days. A simulator, able to process this network is described in Van Landeghem and Debuf (1997). A detailed description of the simulator templates for implementation can be found in Van Landeghem (1998).

As supply chains make use of transport means, their reliability is affected by both variations in demand for transport services and variations in the supply of transport services. Wakabayashi and Iida (1992) define reliability as “the probability of a device performing adequately for the period of time intended under the operating conditions encountered”. The definition is very general in nature and the word ‘adequately’ has to be filled in by the user. In transport, in some cases it might be of primary interest that the journey is finished by a certain time, while in other cases it is more important to assess whether a disruption might be encountered on the journey (Nicholson et al. 2003). Failures appear because the links in the chain are not reliable. Delivery of materials may be late. A distributor may be unable to ship due to weather conditions. In such situations a logistics manager makes use of alternative options. In the former case a safety stock offers a solution. In the latter case alternative routes allow goods to proceed in the chain. In this paper we investigate the construction and verification of model for this kind of supply chains.

SIMULATION OF UNRELIABLE LINKS IN A SUPPLY CHAIN

In order to facilitate the construction of various configurations of a supply chain network with unreliable links we propose a formal procedure. The procedure, if followed strictly, serves as a validation tool because it guarantees an automatic and correct generation of an executable simulation model.

It is stated in Goldratt and Fox (1986) that “Current throughput is in danger” in a JIT system. Failures at any link result in the failure of the entire supply chain, and consequently the overall throughput towards the market is affected. To improve the throughput rate either alternative routes or modes of transport, or intermediate buffer storages can be used.

Similar problems appear in flow line production systems. These production systems consist of a number of stages (arranged in series) at which operations are performed on a workpiece. Operations at the stages are performed by machines or by equipment, which are subject to failure. Also in this case, either stand-by machines or intermediate buffers are used to improve the throughput of the production line.

We use a Petri net as the formal model. The aim is to provide to the designer a set of rules, which generates the places and transitions for the Petri net. The Petri net itself leads to the simulation model. We illustrate the method first for a serial supply chain (i.e. without alternative routes or modes of transport), including safety stock. The operation of each link in the chain can go down for reasons due to its own characteristics. A production operation may fail. A ferry has to stay in the harbour due to a storm. But it can go down due to neighbouring links too. If a previous operation does not provide any raw materials the production can get starved. If the storage at the wholesaler is full, the distribution company is blocked. Depending on the stochastic behaviour of each link and the sizes of the intermediate buffers the performance of the whole chain changes.

The Petri net is further extended with alternative choices. The alternatives are not choices, which are evaluated each time the link is used. They serve to resolve problems if it is likely that the link will get starved, blocked or will cause stoppages further in the chain. Examples are: if a ferry has to stay in the harbour, air carriage can be used; if a supplier is out of stock, goods can be bought from a secondary supplier. It is clear that the use of alternatives influences the chain performance, e.g. in terms of speed or cost.

MODELLING A GLOBAL SUPPLY CHAIN

The behavior of a supply chain with multiple links and intermediate storages is very complex. Logistics managers are interested in performance measures of such a system. *Availability*, or the percentage of time that the chain is supplying to the final customer, is used frequently as an important performance measure. However, analytical results exist only in very simple cases. In other cases either approximations or simulation models are used. If abstracted, the types of models studying a smooth and constant flow of goods from supplier to customer are similar to models of flow-line systems. The name of the systems is borrowed from serial production systems.

In the literature on multi-stage lines with finite intermediate buffers, De Koster (1989) distinguishes four classes of models. His third class deals with continuous flow models. Machine speeds are deterministic but machines may fail. This is the model class, which refers to our type of modeling. Goods flows are deterministic but supply chain links may fail. Some examples of these models can be found in Wijngaard (1979) and in Malathronas et al. (1983).

While obtaining analytical results for a system with many machines is considered to be an impossible task, also the approximative models or simulation models are to be questioned. Approximation models are of two types: aggregation models or decomposition models. The system with two machines and one buffer in a continuous flow-line, has been studied by Malathronas et al. (1983). An approximative model for three machines is formulated by Van Oudheusden and Janssens (1994) in which exponential uptimes and downtimes are assumed for aggregated machines. Both approximation models and simulation models make use of system states. For lines with multiple machines and buffers, the number of states tends to become very large. A formal technique to generate all possible states of the system in an automatic way could be of great help. In the next sections we investigate the opportunities and limitations of a Petri net as a formal modeling tool for flow line systems within the context of a supply chain.

A PETRI NET MODEL FOR A SERIAL SUPPLY CHAIN

A Petri net in its graphical or mathematical form has no physical meaning. However, it is able to model systems in which some events occur concurrently, but in which there are constraints on the concurrence, precedence or frequency of these occurrences. This idea can be used to model supply chains. A place may represent a condition (e.g. a link is available or not), or it can represent a link status (e.g. the number of tokens represents the number of vehicles within the distribution system).

Van der Aalst (1994) has suggested the use of Petri nets for logistics systems. Goods and capacities can be represented by tokens; buffers, storage space and media by places, and operations by transitions. In continuous flow these definitions cannot be held. Individual items do not exist. Tokens and places require other meanings.

As far as we know, only one paper deals with Petri nets used to model automatic transfer lines. Al-Jaar and Desrochers (1990) present two stochastic Petri nets to evaluate the performance of transfer lines. The assumptions are comparable to ours, except for the fact that they work in a discrete production environment instead of a continuous one. Modelling a continuous chain including breakdowns is the main subject of this contribution.

The serial supply chain consists of *n* links, separated by *n-1* storages ($n \geq 2$). The storages in the system can be in one of three states: empty, full or in an intermediary position (labelled 'half'). Links can be either up, down, starved or blocked. A link M is starved if one of the upstream links is down and all storages between this link and link M are empty. A link M is blocked if one of the downstream links is down and all storages between this link and link M are full. When a link is operational and neither starved nor blocked, it supplies goods from the upstream storage to the downstream storage in a continuous way at a constant rate. The situation is shown in Figure 2.

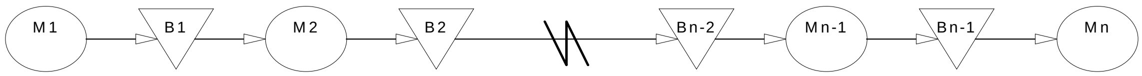

Figure 2: An n-link (n-1)-storage supply chain

States Associated With a Link or a Storage

We propose a Petri net in which a *place* represents the state of a link or of a storage. For each storage, three places are added to the net, indicating that the storage is respectively full, empty or at an intermediate level.

For each link, except the first and the last, four places are added, indicating that the link is respectively up, down, blocked or starved. The first link does not have a place for the state 'starved' and the last one does not have a place for the state 'blocked'. Graphically, the set of places P is represented in Figure 3:

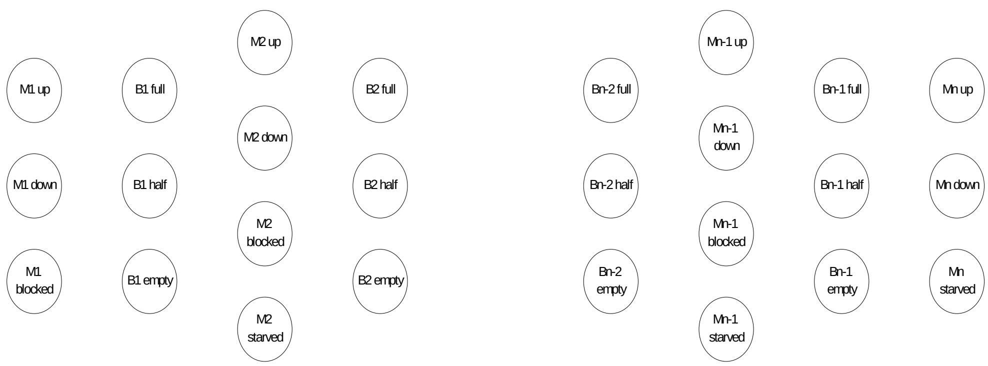

Figure 3: Places of a general Petri net representing the supply chain links and storages

Transitions of the Petri Net Model

Failure or start-up of a link may influence the state of each other link or storage in the chain. Likewise, when a storage becomes empty, this may have an effect on each downstream link or storage. When a storage becomes full, this may have an effect on each upstream link or storage. It is impossible to define a general building block of places and transitions, which can be repeated to describe the behavior of a chain with multiple links and storages.

For example, assume all links are operational, but all storages are empty. If link 1 goes down, all other links become starved. This means that the effect of a single event can spread throughout the entire chain of links and storages and that it is impossible to define a building block of e.g. one link and one storage, which can be repeated to form the Petri net of a serial system with *n*-links and (*n-1*)-storages.

For this reason, we do not concentrate on the transitions of the Petri net, but on the rules to generate the transitions. The transitions model the events, which occur while the system evolves. Four types of events need to be modelled:

1. a link goes down,
2. a link starts up (is repaired),
3. a storage becomes full,
4. a storage becomes empty.

All other changes of state occur as a result of one of these four events. E.g. a link becoming blocked is always the result of a storage becoming full. A storage changing state from full or empty to an intermediary position is always the result of a link failure or a link start-up or of another storage becoming full or empty.

Rule schemata

The rules for generating the Petri net transitions are represented by schemata. A change of state for a specific link or storage is indicated by a square containing the names of two places, separated by the symbol V. Such a square is called a *block*. The upper place is added to the input places of the transition, the lower place is added to its output places. In Figure 4, the breakdown of link *i* is depicted.

The Petri net transition equivalent to this event is also given. The arrow means 'gives rise to the transition'.

As text, the block in Figure 4 can be denoted as [Mi up > Mi down].

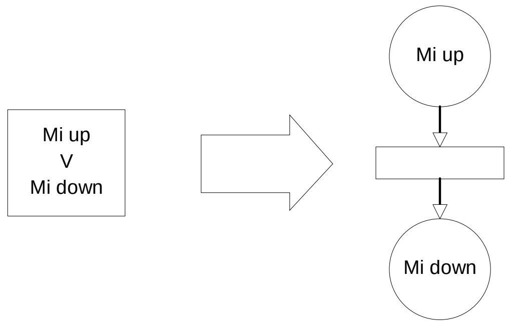

Figure 4: Failure of link i

A block may contain two identical places. This means that there is no change of state for this storage or link, but the places are needed in the transition. Blocks that are adjacent but not connected by arrows can be considered to be one block. All places have to be added to the transition. The details of the rule schemata and the instructions when to use them fall beyond the scope of this paper. A complete description can be found in Sörensen and Janssens (2003).For example, assume we are dressing up the transitions for the event *link 2 starts up* in a 5-link 4-storage system. The schema is given in Figure5.

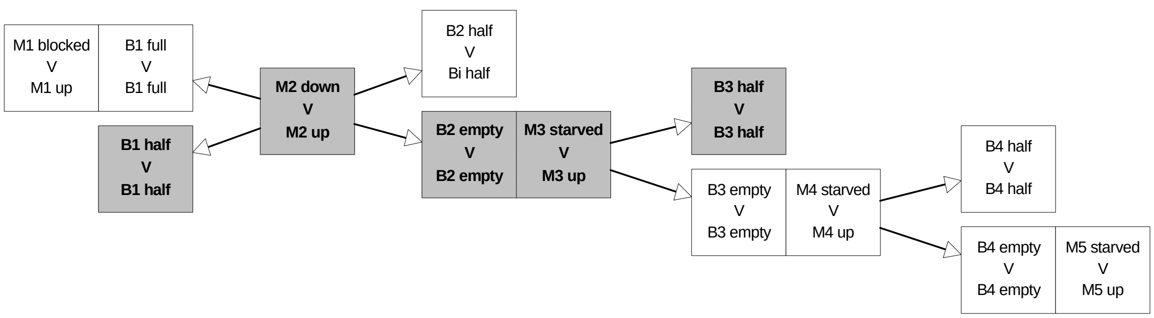

Figure 5: Reduced and expanded schema of link 2 starting up in a 5-link system with one possible path indicated

As an example, one path through the schema is indicated by the shaded blocks. The transition, which belongs to this path is shown in Figure 6

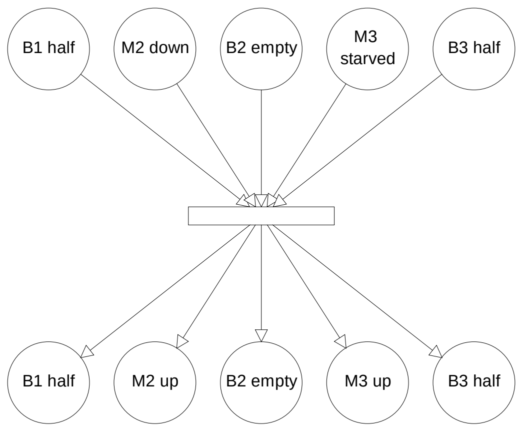

Figure 6: Transition equivalent to the shaded path in figure

There are eight possible paths through the schema in figure 5. Therefore, eight transitions are dressed up for the event *link 2 starts up*.

The Petri net transitions are timeless, i.e. they fire instantaneously, no matter how far-reaching their implications are. Also, transitions cannot fire simultaneously. This excludes some situations from occurring. E.g. If a link is down, the upstream storage cannot be empty, because this situation can only occur if the storage became empty at exactly the same time as the upstream link went down.

TRANSFORMING THE PETRI NET MODEL INTO A SIMULATION MODEL

Token Attributes

To transform the Petri net model into a simulation model, a time aspect needs to be added to the Petri net. This can be reached by attaching one or more attributes to each token in the Petri net. A difference is made between static and dynamic attributes. Static attributes do not alter as the simulation progresses and can be seen as the parameters of the simulation model. Dynamic attributes change during the simulation.

A token belonging to a link invariant (a token that indicates the state of a link) has one dynamic attribute, which has a different meaning depending on the place the token is in (state the link is in). Their meanings are indicated in Table 1.

If the amount of goods in a storage is expressed in a certain volume unit *V* and the progress of time in a time unit *T*, then the speed with which links add or extract goods from the storage is expressed in "volume per time" units (*V/T*). In a balanced chain, we can, without loss of generality, assume that this speed is one for all links.

Link state	Meaning of time property
Up	Time left until the link fails
Down	Time left until the link starts up
Starved / Blocked	Uptime remaining (time left until the link would fail if it were operating)

Table 1: Meaning of an attribute belonging to a link invariant for different link states

Tokens in storage invariants have two attributes. A first one indicates the maximal storage capacity and is of the static type. A second one indicates the current storage content and is of the dynamic type.

Execution of the Simulation

The simulation is now executed according to the logic shown in Figure 7. Steps 1 and 5 are typical in the dynamics of a Petri net model. Step 3 is typical in any discrete event simulation logic. Steps 0, 2 and 4 require some further explanation.

In the explanation following notations are used:

NE	Next-Event time
T_k	Value of simulation clock after the *k*-th transition
$t_{i,k}$	Value of the time property of link *i* after the *k*-th transition
$c_{i,k}$	Value of the storage content property of storage *i* after the *k*-th transition
v_i	Value of the storage capacity property of storage *i*.

Step 0: Initialise The Simulation

Initialising the simulation entails the choice of starting values for the variables in the simulation program. A difference can be made between decision variables (which do not change as the system evolves) and other variables.

Decision variables concern:

- the number of links,
- the capacity of each storage,
- the types of distributions used for failure times and start-up times and their parameters.

Other variables are:

- the state of each machine,
- the content of each buffer.

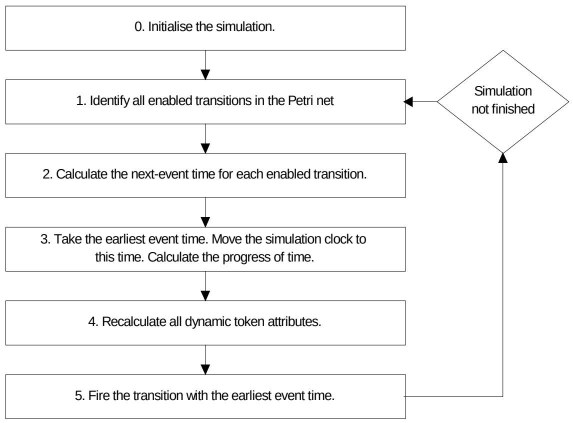

Figure 7: Simulation logic

The initial values of the latter variables should not have an impact on the results of the simulation, providing the simulation runs are long enough.

Step 2: Calculate The Next-event Time For Each Enabled Transition

Depending on the action performed by a transition, the next-event time for this transition is calculated as follows:

Link starts up
The next-event time is calculated by adding the time property of the token (the time remaining until this link goes down) to the current simulation time.
$NE = T_k + t_{i,k}$

Link fails
The next-event time is calculated by adding the time property of the token (the time remaining until this link is repaired) to the current simulation time.
$NE = T_k + t_{i,k}$

Storage becomes empty
The next-event time is calculated by adding the value of the content property to the current simulation time. This can be done because of the assumption that the speed with which links add or extract goods from the buffers equals one.
$NE = T_k + c_{i,k}$

Storage becomes full
The next event time is calculated by adding the difference between the values of the storage capacity and the storage content properties to the current simulation time.
$NE = T_k + (v_i - c_{i,k})$

Step 4: Recalculate all token properties

Because the system of links and storages has evolved for a certain period of time, some of the token properties have changed. Storages have become more full or empty, and links will have gotten closer to their failure or start-up event.
The amount of time, which has elapsed since the last transition is indicated by: ΔT_k ($\Delta T_k = T_{k+1} - T_k$).

Token properties of links, which have gone up or down
If the transition with the earliest next-event time is that of a link going up, the time property of its token is reset and assigned a new random drawing from its Mean Time To Failure (MTTF) distribution. If the transition is that of a link going down, the time property of its token is assigned a new random drawing from its Mean Time To Repair (MTTR) distribution.

Token properties of other links
If the link is up or down, the value of its time property is incremented with the progress in time.
$t_{i,k+1} = t_{i,k} + \Delta T_k$
If the link is starved or blocked, the value of its time property remains unchanged.

Token properties of buffers
For storage *i*, if link *i* is up and link *i+1* is not (it is down or blocked), then the storage is filling up and the content of the storage is increased with the increase in time.
$c_{i,k+1} = c_{i,k} + \Delta T_k$

If link *i+1* is up and link *i* is not (it is down or starved), then the storage is depleted and the content of the storage is decreased with the increase in time.
$c_{i,k+1} = c_{i,k} - \Delta T_k$

If links *i* and *i+1* are both up or both not up, then the value of the content property for storage *i* remains unchanged.

CONCLUSIONS AND EXTENSIONS FOR MORE COMPLEX SUPPLY CHAINS

Supply chain networks seldom show such a simple structure as the one explained in the previous sections. The reader should be aware that a lot of research has to be done before a general and suitable framework for verifying simulation models in this field is worked out completely. We show briefly how the option of alternatives can be embedded into the same framework.

Alternatives have different meanings in a logistics reality. They can refer to a different supplier or shipper, or to an alternative mode of transport. Each link in the chain (called primary link) can have any number of alternative links. They come into operation when a link fails. When the alternative links are up but not operating (they are waiting for the primary or any other link to fail), they are said to be in 'standby mode'. In a real system both the alternatives and the intermediate storages can be combined, as shown in Figure 8.

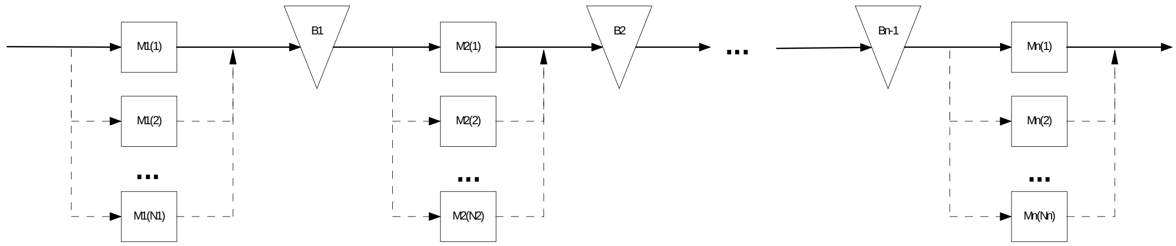

Figure 8: A supply chain network with alternative options and with intermediate storages

Also this type of network can be modeled by means of a Petri net, and its automatic generation is similar. In the case of alternative options, priorities have to be included.

Alternatives have a priority. This means that

- if a link fails, the link with the highest priority of all stand-by alternatives comes into operation.
- if a link with a higher priority than the link currently in operation is repaired, it will start up and interrupt the one currently in operation. The latter is put again in stand-by mode.

Many configurations of supply chain networks can be thought of, but simulation is only of efficient support to the designer if he is sure that the model can be verified. Automatic generation of the simulation model through a formalism, like Petri nets, is certainly of great importance in this field of practice.

REFERENCES

Al-Jaar R.Y. and A.A. Desrochers. 1990. "Performance evaluation of automated manufacturing systems using generalized stochastic Petri nets." *IEEE Transactions on Robotics and Automation*, vol. 6, no. 6, pp. 628-632.

Beamon B. 1998. "Supply chain design and analysis: models and methods." *International Journal of Production Economics*, vol. 55, pp. 281-294.

De Koster M.B.M. 1989. "*Capacity Oriented Analysis and Design of Production Systems.*" Lecture Notes in Economics and Mathematical Systems 323, Springer-Verlag, 245 pp.

Goldratt E.M. and R.E. Fox. 1986. *The Race.* North River Press, Croton-on-Hudson, N.Y.

Malathronas J. P., Perkins J.D. and R.L. Smith. 1983. "The availability of a system of two unreliable machines connected by an intermediate storage tank." *AIIE Transactions*, vol.15, no. 3, pp. 195-201.

Nicholson A., Schmöcker J.-D. and M.G.H. Bell. 2003. "Assessing transport reliability: malevolence and user knowledge." in *The Network Reliability of Transport*, Bell M.G.H. and Iida Y. eds., Pergamon, pp. 1-22.

Slats P.A., Bohla B., Evers J.M. and G. Dijkhuizen. 1995. "Logistic chain modelling." *European Journal of Operational Research*, vol. 87, pp. 1-20.

Sörensen K. and G.K. Janssens. 2003. "A Petri net model for a continuous flow transfer line with unreliable machines." *European Journal of Operational Research*, vol. 151, pp. 253-264.

Van der Aalst W.M.P. 1994. "Putting high-level Petri nets to work in industry." *Computers in Industry*, vol. 25 no. 1, pp. 45-54.

Van Landeghem R. 1998. "Experiments with MISTRAL, a supply chain simulator." in *Proceecings of the International Workshop on Modelling and Simulation within a Maritime Environment.* September 1998, Riga, Latvia, pp. 71-75.

Van Landeghem R. and M. Debuf. 1997. "Supply chain characterisation through Monte Carlo simulation." In *Proceedings of the Production Economics Conference.* July 1997, Göteborg, Sweden

Van Oudheusden D.L. and G.K. Janssens. 1994. "Availability of three-machine two-buffer systems." *JORBEL (Belgian Journal of Operations Research, Statistics and Computer Science)*, vol. 34, no. 2, pp. 17-37.

Vidal C.J. and M. Goetschalckx. 1996. "The role and limitations of quantitative techniques in the strategic design of global logistics systems." School of

Industrial and Systems Engineering Research Report, Georgia Institute of Technology, Atlanta, GA.

Wijngaard J. 1979. “The effect of inter-stage buffer storage on the output of two unreliable production units in series, with different production rates.” *AIIE Transactions*, vol. 11, no. 1, pp. 42-47.

AUTHOR BIOGRAPHIES

Gerrit K. Janssens (°1956) received degrees of M.Sc. in Engineering with Economy from the University of Antwerp (RUCA), Belgium, M.Sc. in Computer Science from the University of Ghent (RUG), Belgium, and Ph.D. from the Free University of Brussels (VUB), Belgium. After some years of work at General Motors Continental, Antwerp, he joined the University of Antwerp until the year 2000. Currently he is Professor of Operations Management and Logistics at Hasselt University within the Faculty of Business Administration. He also holds the CPIM certificate of the American Production and Inventory Control Society (APICS). During the last fifteen years he has repeatedly been visiting faculty member of universities in Thailand, Vietnam, Cambodia and Zimbabwe. His main research interests include the development and application of operations research models in production and distribution logistics.

Katrien Ramaekers (°1979) graduated as Master of Business Economics, option Technology at the Limburg University Centre in 2002. In October 2002, she started as a Ph.D.-student at Hasselt University. Her main research interest is on the integration of simulation and optimisation, especially as a support for complex logistics decision-making and for decision support with limited information in supply and demand. She is a member of the research group Data Analysis and Modelling and the Transportation Research Institute of Hasselt University.

Kenneth Sörensen holds a degree of Commercial Engineer in Information Systems from the University of Antwerp, RUCA (1997). He obtained his PhD in Applied Economics from the University of Antwerp in June 2003. His main research interests include optimization using metaheuristics, supply chain design and the design of production systems. He has published several papers in international journals on these topics. He is also founder and coordinator of the EURO working group on metaheuristics EU/ME.

Rik Van Landeghem holds a degree of Civil Engineer in Electro-Mechanics (RUG)and a MSc in Industrial Management (RUG). He obtained his PhD at the Ghent University in 1984. For several years he worked as a manufacturing consultant in the field of industrial informatics. He joined the BMT group in 1989, where he successively became EDP manager, Plant manager of the Sheet Metal division and finally MIS Director. Since 1993 he is a full time professor at the Ghent University, as well as Head of the Department of Industrial Management (Faculty of Applied Sciences). He is also director of the CIM-CIL Technology Transfer Center and member of the board of PICS Belgium. He is an expert in logistics systems and their application in industry. Other research interests include Concurrent Engineering. He is editor of a casebook on Computer Integrated Manufacturing, as well as co-author of a book on ERP Implementations.