

SCOlog: A logic-based approach to analysing supply chain operation dynamics

Areti Manataki, Yun-Heh Chen-Burger, Michael Rovatsos

*School of Informatics, The University of Edinburgh,
Informatics Forum, 10 Crichton Street, Edinburgh EH8 9AB, UK*

*A.Manataki@sms.ed.ac.uk, jessicac@inf.ed.ac.uk,
mrovatso@inf.ed.ac.uk*

Abstract

In a complex business world, characterised by globalisation and rapid rhythms of change, understanding supply chain (SC) operation dynamics is crucial. This paper describes a logic-based approach to analysing SC operation dynamics, named SCOlog. SC operation is modelled in a declarative fashion and it is simulated following rule-based execution semantics. This approach facilitates the automated explanation of simulated SC operational behaviours and performance. The automated explanation support provided by SCOlog is found to improve the understanding of the domain for non-SCM experts. Furthermore, SCOlog allows for maintainability and reusability.

Keywords: supply chain modelling, supply chain simulation, supply chain operation, supply chain dynamics, knowledge based system, workflow, business rules, intelligent agent

1. Introduction

The modern business landscape is highly dynamic and competitive. The current boundaryless corporate arena is characterised by rapid rhythms of change and a high degree of uncertainty. In order to succeed in the 21st century global and volatile market, companies can no longer compete in isolation from their Supply Chain (SC) partners. SC- rather than enterprise-based competition is now experienced (Harrison and van Hoek, 2008), and thus “supply chain management consciousness is accelerating up the corporate agenda” (Storey et al. 2006, p.757).

One of the most prominent problems in Supply Chain Management (SCM) involves the end-to-end integration of supply chains (Gunasekaran and Ngai, 2004). This involves aligning objectives with upstream and downstream SC partners, streamlining SC operations and coordinating activities across the supply chain. In today’s distributed and uncertain environment there is an imperative requirement for SC integration (Butner, 2010). At the same time, the emergence of global SC operations poses a great challenge to integrating supply chains and increases SC risk. Achieving resilient and fully integrated supply chains is a demanding task that requires a deep understanding of supply chain management dynamics.

Although the SCM research community has long recognised the significance of analysing SCM dynamics (Lee et al. 1997; Choi et al. 2001; Sarimveis et al. 2008), relatively little has been reported on the matter of analysing SC operation dynamics. Hence, the interrelationships between the operational behaviour (i.e. decisions, actions and interactions) of SC members during SC operation have been overlooked by existing literature, despite the fact that understanding SC operation dynamics is essential for coordinating SC activities and integrating supply chains. First, companies need to understand the interdependencies between different aspects of their SC operational behaviours in order to achieve internal integration. For example, how do their every-day sourcing decisions and actions affect their production activities? Such information is crucial for streamlining and synchronising the operation of separate business functions. Second, SC managers need to have a deep understanding of the interrelationships between the operational behaviour of different SC members; this is particularly important when employing upstream and downstream SC integration initiatives, as SC-wide process transparency is a critical antecedent to effective coordination of SC activities (Fawcett and Magnan, 2002). This involves, for instance, understanding how the production decision-making of an organisation at the

very first tiers of the supply chain affects the sourcing activities of a company within the last SC tiers. Third, the effect of individual SC members' operational behaviour on SC-wide performance should be clear to SCM practitioners. Such knowledge could help towards repairing problematic SC operation and improving SC performance. Fourth, the interrelationships between SC members' operational behaviour and SC disruptions should be taken into account when planning or implementing SC integration projects. This involves identifying the effect of disruptive events on individual SC members, and how this may be propagated along the supply chain. It also includes discovering whether specific disruptions occur due to the operational behaviour of certain SC members. The above illustrate the significance of understanding SC operation dynamics and motivate our research.

The interdependencies described above are often non-obvious, and discovering them is a challenging task. The problem becomes even harder when one considers the distributed, dynamic and complex nature of modern supply chains. Artificial intelligence and knowledge-based techniques (Stefik, 1995) are well-known for analysing complex and dynamic systems, and providing rigorous decision support. More relatively, they can automatically derive transparent explanations of complex behaviours. They also enable the explicit diagnosis of problematic situations, which can be supported by valuable explanations.

This paper employs artificial intelligence techniques for analysing supply chain operation dynamics. In accordance with the four aspects of SC operation dynamics described above, we focus on identifying the interrelationships that lie:

- between different aspects of an SC member's operational behaviour
- between the operational behaviour of different SC members
- between SC members' operational behaviour and SC performance
- between SC members' operational behaviour and SC disruptions

We propose a logic- and knowledge-based approach to the research problem, named 'SCOlog' (Supply Chain Operation dynamics explained through a LOGic-based approach). SCOlog consists of three components: First, a framework for modelling SC operation is proposed, covering commonly agreed aspects of SC operation, while recognising current trends and issues of the field. The modelling constructs are declaratively specified and they can be directly used to model SC

operation scenarios. Second, rule-based execution semantics of this model are specified, based on which a simulation environment is implemented. The developed system can be used to simulate complex SC operation scenarios and perform what-if analysis. Third, a mechanism for generating explanations of simulated SC operation is designed by utilising the declarative formalism of SC operation constructs and the rule-based specification of execution semantics. An explanation system is implemented which can automatically answer questions on SC operation dynamics.

We hypothesise that *SCOlog generates explanations that improve the understanding of the domain for non-SCM experts and employs a logic-based approach to the modelling and simulation of supply chain operation, allowing for maintainability and reusability*. This hypothesis is decomposed in two research claims:

1. The use of automated explanation support improves the understanding of the domain for non-SCM experts, with respect to their (a) time-efficiency and (b) correctness when explaining SC operation dynamics. The correctness improvement is bigger compared to the case where no automated explanation support is available, without loss of time-efficiency.
2. A logic-based approach for modelling, simulating and explaining SC operation scenarios allows for maintainability and reusability with respect to (a) the specified SC operation input models, (b) the developed simulation system and (c) the developed explanation system.

This paper is organised in six sections. Section 2 provides background knowledge on SCM and artificial intelligence technologies relevant to *SCOlog*, and it discusses related work in the area of SC simulation. Section 3 presents the proposed framework for modelling, simulating and explaining SC operation. With the use of a supply chain scenario, Section 4 demonstrates the value of *SCOlog*. Section 5 evaluates the research claims with the use of empirical and analytical methods. The paper concludes with a discussion of the implications and limitations of *SCOlog*.

2. Literature Review

2.1 Supply chain management

A Supply Chain “consists of all parties involved, directly or indirectly, in fulfilling a customer request” (Chopra and Meindl 2003, p.4). During SC operation

products, funds and information flow across the supply chain. Supply Chain Management involves managing these flows in order to maximise total supply chain profitability (Chopra and Meindl, 2003, p.6). Lately there is a shift from the antagonistic to a collaborative SCM model (Storey et al. 2006), thus there is a requirement for the full alignment and integration of supply chains. As recently stated by Stock et al. (2010, p.36), “the issue of how to integrate multiple organisations into one cohesive supply chain is an important one to investigate”. Although SCM scholars promote a holistic view of the field, most of the existing research still focuses on specific SC links or nodes (Giunipero et al. 2008). Moreover, SCM practice is still far from the vision of SC integration, as supply chains often fail to behave as one entity (Holweg and Pil, 2008).

There are different approaches in literature towards SC integration. A considerable stream of research advocates information sharing between SC members as a means to integrate supply chains (Lee et al. 2000; Yu et al. 2001; Fiala, 2005). The role of information technology has been instrumental for employing information sharing initiatives (Gunasekaran and Ngai, 2004; Rai et al. 2006), and recent advances such as EDI, ERP systems and RFID have been widely used in this context. A different approach to integrating supply chains involves strategic collaboration between SC partners (Barratt, 2004; Holweg et al. 2005). This includes partnerships and strategic alliances between SC members, collaborative planning, forecasting and replenishment throughout the supply chain, as well as practices like vendor managed inventory and continuous replenishment programs. Another part of the research community recognises the importance of coordinating activities across the supply chain in order to achieve SC-wide integration (Simatupang et al. 2002; Lee and Whang, 2004; Arshinder et al. 2008; Frohlich and Westbrook, 2001; Fawcett and Magnan, 2002; Barua et al. 2004). This involves streamlining and synchronising SCM actions along the supply chain, such as procurement decisions, transportation and order fulfilment activities. The tight coupling between activities across the supply chain not only helps towards SC integration, but it can also bring efficiency benefits in terms of accuracy, time and cost. Despite the evident importance of SC coordination for integrating supply chains, it has been argued that the majority of SC coordination efforts concentrate on intra- rather inter-organisational supply chains (Stadtler, 2008). A reason behind this might be the difficulty of understanding SC

operation dynamics, which is a prerequisite for coordinating SC activities (Fawcett and Magnan, 2002; Chan and Chan, 2010; Puigjaner and Laínez, 2008).

Understanding SCM dynamics is one of the most critical issues in SCM research and practice. There is an extensive body of work on SCM dynamics in the context of SC planning and demand forecasting (Lee et al. 1997; Riddalls et al. 2000; Hwarng and Xie, 2008). Research in this area focuses mostly on the bullwhip effect, which is the phenomenon of the amplification of demand order variability as we move up in the supply chain. Typical approaches to this problem include differential equation models and system dynamics simulation. Studies have also appeared on the matter of SC configuration considering aspects of SCM dynamics (Akkermans, 2001; Choi et al. 2001). This stream of research deals with specifying the SC system's structure, policies and processes, while taking into account the interdependencies between the decisions of different SC members. Representative methods employed in this field include complex adaptive systems and intelligent agents. While there is a considerable research effort to deal with SCM dynamics in the context of SC planning, demand forecasting and SC configuration, this is not the case for the context of SC operation. Hence, the problem of analysing SC operation dynamics remains understudied.

SC operation dynamics refer to the interrelationships between the operational behaviour of SC members. During SC operation, SC members make decisions, act and interact, leading to the flow of products, funds and information across the supply chain. The decisions, actions and interactions of individual SC members have a direct effect on the operational behaviour of other SC members, thus influencing the supply chain as a whole. However, these interdependencies are not clear to SC managers, thus hampering their efforts towards coordinating SC activities. As Barratt (2004) observes, the effect of external processes on an organisation's internal processes is often neglected. At the same time, even experienced SC managers ignore the long-term consequences of their actions (Serman, 2006). SCM practitioners often adopt suboptimal mental models, since "cause and effect are obscure, creating ambiguity and uncertainty" (Serman, 2006, p.34). As already mentioned, SC operation dynamics have been overlooked by existing literature. To our knowledge, the only research area that captures, to some extent, SC operation dynamics is that of SC simulation. Related work in SC simulation is more thoroughly discussed in Section 2.3, but it is worth mentioning that it does not explicitly address SC operation dynamics. This means that during SC simulation the existence of SC operation

dynamics can be observed, but it is not explained or analysed. This is a considerable gap, as there is a need for understanding and explaining complex SCM phenomena (i.e. finding reasons for their existence and discovering underlying generative mechanisms) rather than simply describing them (Adamides et al. 2012).

When studying SC operation dynamics, one should take into account two parameters that affect SC operation. Firstly, agility should be considered, which is “the ability of an organisation to respond rapidly to changes in demand” (Christopher 2000, p.38). Agility is crucial in the current business landscape, characterised by rapid rhythms of change and high degree of uncertainty. SC agility is also known to support SC responsiveness (Gunasekaran et al. 2008) and SC resilience (Christopher and Peck, 2004), which involves reacting to SC disruptions in a flexible and adaptive manner. Secondly, SC disruptions should be taken into consideration. SC disruptions are “unplanned and unanticipated events that disrupt the normal flow of goods and materials within a supply chain” (Craighead et al. 2007). The occurrence of disruptive events and the resulting poor performance are increasingly common, mainly due to SC globalisation and the wide use of outsourcing practices. Hence, managing SC disruptions is perceived as one of the most important current and future issues of the field (Butner, 2010; Melnyk et al. 2009). In order to effectively manage SC disruptions, one needs to understand the interrelationships between disruptions and SC operation, but according to Blackhurst et al. (2005) there is limited research towards understanding the impact of disruptions on the SC system.

2.2 Artificial intelligence technologies for supply chain management

Artificial intelligence techniques have shown great potential in supporting and improving human decision-making for complex problems, where optimal solutions are difficult to produce. They provide transparent and rigorous reasoning mechanisms that allow the capturing and explanation of complex behaviours, like the ones exhibited in a supply chain management context. Relevant technologies that have been applied to SCM problems include knowledge-based systems, agents and intelligent workflows.

A knowledge-based system (KBS) is “a computer system that represents and uses knowledge to carry out a task” (Stefik, 1995). KBSs can diagnose, prognose and explain complex problems. Knowledge-based techniques can also drive and support simulation in two ways. Firstly, they can explain simulation behaviours and results to

the user; this is particularly useful in the case of complex and dynamic systems, where simulation results are non-obvious. Secondly, they enable decision-making at runtime; this is valuable for dynamic domains, where adaptive and flexible behaviours are common. Given the complex, dynamic and flexible nature of SCM, we regard KB-simulation as highly relevant to the problem of analysing SC operation dynamics. However, there is a scarcity of relevant research efforts in existing literature.

Intelligent agents are computer systems situated in some environment, upon which they can autonomously act in order to meet their design objectives (Wooldridge 2002). Agent technologies can tackle SC planning and demand forecasting (Fox et al. 2000) and SC configuration problems (Piramuthu, 2005). Furthermore, Swaminathan et al. (1998) suggest an agent-oriented modelling framework for simulating SC operation. Their work addresses all flows across the supply chain, but it does not shed any light on how the activities of a single SC member affect other SC members.

Business process modelling and workflow management systems (van der Aalst and Stahl, 2011) define, manage and execute workflows, thus supporting the automation and analysis of organisational procedures. Given that SCM is widely understood in terms of processes (Burgess et al. 2006), workflow technologies are highly relevant to the management of SC operations. Although workflow techniques have been successfully applied to SCM problems (Liu et al. 2005; Goutsos and Karacapilidis, 2004), the issue of analysing SC operation dynamics remains unexplored.

2.3 Related work in supply chain simulation

The problem of analysing supply chain operation dynamics has not been thoroughly addressed by existing literature so far. Nevertheless, we recognise that the area of simulation can capture, to some degree, SCM dynamics. SC simulation is relevant to the studied problem, as it provides an insight into SC-wide operation and allows the analysis of SC performance for different scenarios. There is a plethora of off-the-shelf SC simulators, including SC Analyzer, Supply Chain Guru and SmartSCOR. IBM SC Analyzer (Bagchi et al. 1998; Archibald et al. 1999) combines optimisation and simulation techniques to analyse issues such as site location, manufacturing and transportation policies, as well as customer service. The following seven SC roles and functions can be modelled and simulated: customer,

manufacturing, distribution, transportation, inventory planning, forecasting and supply planning. The tool's outputs involve cost, as well as fill rates, return rates, etc.

Llamasoft Supply Chain Guru (LlamaSoft Incorporated, 2012) is another software tool that combines optimisation and simulation. Its simulation component serves mainly as a validation of the proposed optimal SC design, and it can be used to predict and test the effects of the suggested SCM changes. The basic elements of a Supply Chain Guru model are the following: products, sites, demand, sourcing policies, transportation policies and inventory policies. Simulation output includes financial reports, inventory units, customer service rates and resource utilisation rates, which are visualised in sum-statistics and time series graphs.

IBM SmartSCOR (Dong et al. 2006; Ren et al. 2010) is a supply chain transformation platform that employs mixed integer programming techniques and process-oriented simulation and analysis. The basic elements of a simulation input model are entities, products, resources and processes. Simulation in SmartSCOR is driven by IBM's WebSphere Business Modeller, allowing for rich static and dynamic analysis. Additionally, SmartSCOR facilitates so-called root cause analysis, which in this case does not involve automated diagnosis, but instead the use of fishbone diagrams by business experts in order to assist them with the qualitative identification of root causes. Hence, even though SmartSCOR recognises the need and usefulness of causal analysis for SC operation, the support it provides is limited.

There is also a growing body of research in SC simulation. Stefanovic et al. (2009) develop an SC simulation environment by adopting a process-oriented approach that utilises the SCOR model. They identify four components of an SC model: supply network structure, process, business environment and constraints submodel. The main component of the developed simulation software is a database that contains a process library and a collection of previously defined simulation models; this approach facilitates the process of specifying simulation input and allows the storage and querying of simulation results of different scenarios. However, the capabilities of this querying are not made clear in this paper, and the analysis of simulation results is not thoroughly discussed.

Longo and Mirabelli (2008) adopt a data-oriented approach for simulating supply chains, and they demonstrate it with the use of the simulation environment eM-Plant. Their SC simulation model consists of stores, plants and distribution centres, as well as inventory policies. The simulation output includes SC performance metrics such as

fill rates, on hand inventory and inventory costs. For experimentation with different scenarios and what-if analysis, the authors propose the use of the simulator jointly with appropriate design of experiments and analysis of variance.

SCOR template (Persson and Araldi, 2009; Persson, 2011) is a set of SCOR-based building blocks in the general-purpose simulation software Arena. The objective of this research effort is to ease the process of specifying SC simulation input models for SCM practitioners. In order to achieve this, the authors utilise SCOR processes and metrics to define appropriate modules in Arena, which can be directly used by supply chain managers.

2.3.1 Current limitations

Existing SC simulation approaches have considerable strengths, especially with respect to usability. However, three main limitations are identified, which are important for analysing SC operation dynamics:

1. SC simulation results are not explained, and simulation is treated as a black box. This means that it is not possible to obtain answers on SC performance and SC operational behaviours in an automated way. This is a considerable gap given the highly complex operation of modern supply chains.
2. SC disruptions are not analysed, and often they are not explicitly modelled. This means two things. Firstly, simulated SC behaviours and performance are not linked to the occurrence of disruptions. Secondly, the propagation of SC disruptions is not investigated, and the effect of SC disruptions on SC operation is not made clear.
3. SC agility aspects are typically not incorporated in SC simulation models. This means that it is not possible to model and simulate highly flexible operations or decision-making; as a consequence, agile behaviours cannot be explicitly analysed as part of SC operation.

We regard the first limitation to be the primary one with respect to the studied problem. Discovering interdependencies between the operational behaviour of SC members is a challenging task when simulating the operation of complex supply chains. Given the lack of explanation of simulation results, it is hard to manually analyse any of the four aspects of SC operation dynamics, as identified in Section 1. Interrelationships between different aspects of SC operational behaviour or between

the behaviours of different SC members are not obvious. Similarly, determining causes and effects of low SC performance based only on simulation results is a demanding process. At the same time, the second limitation implies that specifying reasons or consequences of SC disruptions is not supported by existing work. The third limitation suggests that flexible behaviours cannot be studied, even though agility is a hot topic in SCM. These limitations demonstrate that existing work in SC simulation only touches the surface of the studied problem and fails to target its core.

3. SColog: A knowledge-based approach to modelling, simulating and explaining SC operation dynamics

We propose a knowledge-based approach to the problem of analysing SC operation dynamics. The devised solution addresses the three limitations of existing related work, a matter that is highlighted throughout this section. We begin by introducing the conceptualisation of SC operation and its declarative formalisation. The simulation of SC operation in a rule-based fashion is then discussed. Finally, a mechanism for automatically generating explanations of simulated SC operation is presented.

3.1 Modelling SC operation

We identify appropriate constructs for conceptualising SC operation and we declaratively specify them through Prolog-based predicates. These constructs are classified into three categories: (1) structural constructs, which are things that exist in an SC and that are highly relevant to SC operation dynamics, (2) behavioural constructs, which describe the operational SCM behaviour of SC members and (3) disruption-related constructs, which specialise on problematic SC operation.

3.1.1 Structural constructs of SC operation

There are six main types of structural SC constructs: SC members, products, resources, funds, information and events. *SC members* are the main actors of the SC, and their behaviour drives SC operation. SC members are technically specified through intelligent agents. There are two main reasons behind this decision. Firstly, intelligent agents' characteristics of autonomy, social ability, reactivity and proactiveness are highly relevant to SC members' behaviour during SC operation. Secondly, an agent-oriented view of SC operation allows its study at two levels: the SC member-specific and the global of the SC; this is particularly useful for analysing

SC operation dynamics. The predicate-based definition of agent-oriented SC members is provided below, uniquely identifying SC members. SC agents are further described through behavioural constructs, which are discussed in Section 3.1.2.

```
supply_chain_member(AgentId)
```

The flow of *products* is the most important aspect of SC operation, while *resources* (e.g. equipment or machinery) support SC operation. Products and resources are entities that exist at some SC member at a certain timepoint, and they thus belong to the corresponding agent's local environment. Their definition is entity-oriented and does not explicitly distinguish between products and resources, allowing for economy when implementing the simulation environment. In the case of products, inventory levels, safety stock and bills of materials are also specified. The predicate-based definition of entities and inventory is provided below.

```
entity_occ(AgentId, EntityName, EntityId)
inventory(AgentId, Status, EntityName, EntityAmount, ListOfEntityIds)
```

Funds flow across the SC (upstream) in return for the downstream flow of products. Their availability at some SC member is a prerequisite for the SC member's operational behaviour. The declarative specification of the level of funds at some SC member at a certain timepoint follows.

```
funds(AgentId, FundsCategory, FundsAmount, ListOfOrderIds)
```

Information is available and can be exchanged between SC members to support SC operation. It covers subjects such as orders, lot sizes and SC partners. The source of information at some SC member can be the SC member itself or other SC members. Information that is created locally by the SC member can be generically specified as data, or it can be specified in a specialised manner (i.e. through specialised predicates for placed and received orders, scheduled production and transportation requests). Information that is received by other SC members is specified in the form of facts. Relevant predicate-based definitions are provided below.

```
data(AgentId, SubjectID, Content)
placed_order(OrderId, AgentId, OrderingToAgentId, DestinationAgentId,
  EntityName, EntityAmount, ScheduledReceiptTime, ActualReceiptTime)
fact(AgentId, Content)
```

Events are incidents that can be the triggers but also the consequences of SC operation. They can occur at the global or the local SC level and they may be internally or externally created. The formal representation of events is provided below. Note that *InvokerId* refers to the invoker of the event occurrence, while *EventFlag* links the event occurrence to a specific SCM operation.

```
event(AgentId, EventId, EventName, EventFlag, InvokerId, T)
```

3.1.2 Behavioural constructs of SC operation

We identify three facets of SC members' operational behaviour: thinking, acting and interacting. Thinking refers to the decision-making process of SC members on operational matters. It may involve standard, routine decisions (e.g. when to place an order) or flexibility decisions (e.g. how to react to machinery breakage). Acting is the most important aspect of SC members' operational behaviour, and it refers to their extrinsic behaviour that causes the flow of products, funds and information across the SC. The SCOR model (Supply Chain Council, 2008) is adopted for conceptualising SC members' acting behaviour, as it is a widely accepted reference model of SC operation (Bolstorff and Rosenbaum, 2012). This way, four areas of operational acting for each SC member are recognised (i.e. source, make, deliver and return), with a focus on execution. Interacting refers to communication between SC members through the exchange of messages. SC members may communicate as part of their standard order management behaviour or in order to deal with unexpected situations.

Mapping this conceptualisation to an agent-oriented representation, we regard each SC member as an intelligent agent consisting of three layers:

- **Reasoning layer:** corresponds to the agent's ability to think and make decisions
- **Process layer:** corresponds to the agent's ability to execute processes, and thus act upon the environment
- **Communication layer:** corresponds to the agent's ability to receive and send messages, and thus interact with other agents

An agent's reasoning layer is represented through *Business Rules* (The Business Rules Group, 2000). Business Rules (BRs) can describe various types of principles that guide SC reasoning at different levels of detail and complexity. We recognise three types of BRs in the context of SC operation: (1) policies, (2) flexibility BRs and

(3) process preconditions. A generic, declarative specification of a BR at some SC member is provided below.

```
br(AgentId, BrID, BrType, BrContent)
```

The general form of a BR's content is: `ifthen(IFpart, THENpart)`, where `IFpart` expresses the conditions of the BR, and `THENpart` its consequences. `IFpart` is a declarative expression, consisting of conjunctions and/or disjunctions of predicates, and it can be highly complex, if needed. `THENpart` is a list of consequences, which can be of reasoning, acting or interacting nature. BRs for time- and quantity-based policies follow this formalism, while a specialised representation is adopted for popular ordering policies, such as the (R,Q) and the (s,S) policies (Axsäter, 2006). The generic formalism is also followed for flexibility BRs: `THENpart` defines the reaction to the problematic situation described through `IFpart`. We consider the explicit specification of flexibility business rules as a strength of this modelling framework, as this way SC agility aspects are incorporated. Business rules that serve as process preconditions follow the `br/4` specification, and their `BrContent` consists of one predicate that expresses a process precondition.

An agent's process layer is represented through *Business Processes*. There are three main reasons behind this decision. First, SC members' acting is conceptualised based on SCOR model's processes, which are naturally formalised through BPs. Second, BPs are suitable for capturing aspects of SC operational dynamics, given that their preconditions and postconditions are formally specified. Third, BP decomposition allows for description of SC members' acting behaviour at different levels of detail. We recognise the Fundamental Business Process Modelling Language (FBPML) (Chen-Burger et al. 2002) as a useful foundation for formalising SC business processes, as it has formal semantics, it allows for the description of business process models with complex structure, and it facilitates their translation into executable workflows. The definitions presented in this section are an extension of previous work (Manataki and Chen-Burger, 2009) that followed FBPML. The declarative, predicate-based specification of a BP at some SC member is provided below. A process is executed if its preconditions and trigger conditions hold. A trigger is an event that invokes process execution, while a precondition is a requirement for process execution which makes sure that its actions can be carried out successfully by the agent. Preconditions involve the availability of entities, funds and information at

some SC member, while there are also BR-based preconditions. The execution of a process brings about the performance of the actions defined in `ActionList`, thus modifying the world state. Actions transform, create or delete entities, funds and information, and they cause the occurrence of events. The execution semantics of the formalised BPs are further discussed in Section 3.2.

```
process(AgentId, Pid, PName, TriggerList, PreconditionList,
       ActionList, Duration, Cost)
```

The formal model of an SC agent's process layer also includes the junctions in the involved business process model (BPM), thus describing the control sequence of the BPs in the BPM. The following FBPML-based junction types are considered: start, finish, link, and-joint, or-joint, and-split and or-split. The predicate-based specification of a junction follows, where `JType` refers to the junction type, `PreList` is the list of processes or junctions that are directly preceding the junction, and `PostList` is the list of processes or junctions that are directly following it.

```
junction(AgentId, Jid, JType, PreList, PostList)
```

An agent's communication layer is represented through *communicative actions*, which involve sending and receiving messages. The declarative specification of messages is provided below, where `Sender` refers to the agent that sends the message and `ReceiversList` refers to the agents to which the message is addressed. A message can be a reply to a previous message (as denoted at `InReplyTo`), and it can be characterised by a Performative such as inform, refuse, propose, etc.

```
message(MessageID, Sender, ReceiversList, InReplyTo, Performative,
       Content, T)
```

Apart from the three-layered definition of SC agents' operational behaviour, we also identify behavioural meta-constructs on *SC performance*. We use the SCOR-based framework for SC performance measurement (Supply Chain Council, 2008), as it specifies the calculation of a wide range of metrics, while linking them to involved processes. Moreover, given its hierarchical structure, it is easy to use in the context of large SCs with complex operations. We recognise performance metrics along four SC performance attributes: reliability, responsiveness, cost and asset management. The formalisation of SC performance metrics follows the general form of `performance_metric(AgentId, Value)`. An illustrative example follows:

```
cycle_time(AgentId, source, Product, Value)
```

3.1.3 Constructs for problematic SC operation

We conceptualise problematic SC operation with respect to product flow, and we identify two dimensions: problematic situations that arise during SC operation and low SC performance. As far as the first is concerned, five types of *problematic situations* are identified: First, delays can occur at some SC member. These delays can involve any SCOR-based operational acting area and they may refer to the long duration of some process, its late starting or its late completion. Taking these two dimensions into account, we can have source-start delays, make-finish delays, deliver-duration delays, etc. Second, quality issues can arise at some SC member, involving either resources or products that are available. Such examples are machine breakdowns, product damages and errors with items that lead to their destruction. Third, SC members can act unusually, possibly as a result of flexibility decisions that they make in the case of problematic situations. Such an example is the urgent sourcing from a non-standard supplier. Fourth, demand fluctuation can take place, a typical example of which is the receipt of unusually big orders. Fifth, order deliveries can be cancelled, causing trouble to the SC member awaiting the order. Categorising these five types of problematic situations based on their source, the first three are experienced internally, the fourth is experienced through the demand side and the fifth through the supply side. These problematic situations are declaratively specified through Prolog-based predicates. An example for process start delays follows, where `ProcessInst` refers to a particular instance of an SC agent's process that is executed.

```
process_start_delay(ProcessInst)
```

As far as *low SC performance* is concerned, this may involve any of the SCOR-based performance metrics. SC performance is understood as low when the actual values of the metrics are beyond some threshold defined by the SC or the corresponding SC member. For reasons of simplicity, we focus on the following subset of cases of low SC performance: high cost, high cycle times, low on time rates.

3.2 Simulating SC operation

In this section we present the adopted framework for simulating SC-wide operation and we discuss aspects of an appropriately implemented simulation environment. Our aim is to fill the three gaps identified in existing SC simulation solutions, as presented in Section 2. In order to fill the first gap we adopt a knowledge-based approach, so as to enable the automated generation of explanations

of SC operation dynamics. A mechanism for detecting SC disruptions is also provided, thus addressing the second gap. As far as the third gap is concerned, decision-making for agility purposes is simulated with the use of a reasoning engine.

3.2.1 Simulation system design and architecture

The main simulation *input* is the formal model of the operation of a supply chain, as described in Section 3.1. There are three categories of simulation *output*: (1) real-time SC operation, (2) measured SC performance and (3) detected problematic situations. Measured SC performance and the detected problematic SC operation involve aspects discussed in Section 3.1.

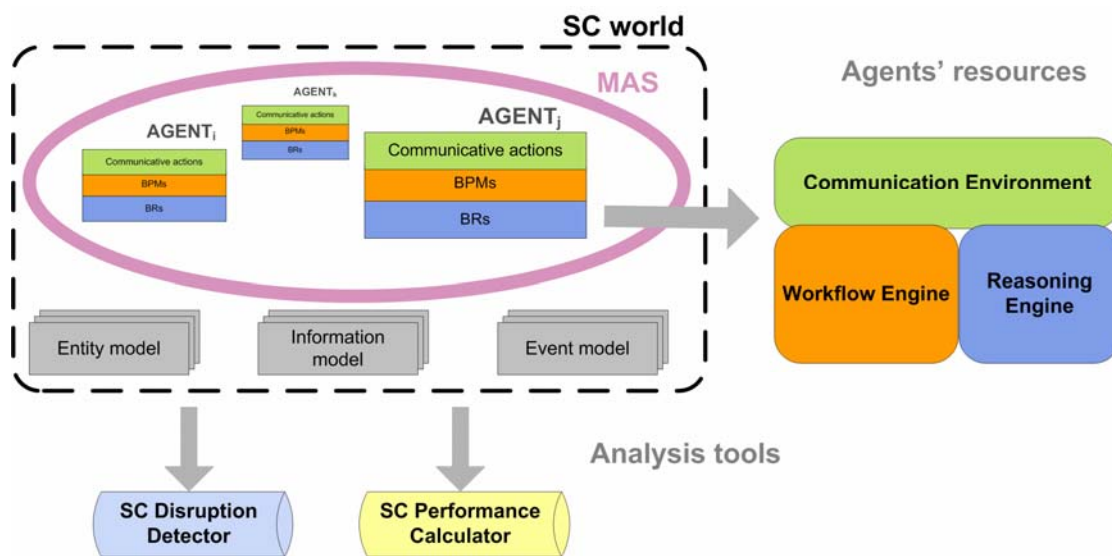


Fig. 1: Simulation system architecture

The *architecture* of the simulation system is presented in Figure 1, where three main components can be seen: SC world, agents' resources and analysis tools. The SC world consists of an SC multiagent system, the entities and information available and the SC events that occur. An SC agent consists of three layers, as discussed in Section 3.1.2: BRs, BPMs and communication capabilities. In order to exhibit dynamic behaviour, an SC agent uses resources that drive SC simulation. The resources that are available to SC agents are: a workflow engine, a reasoning engine and a communication environment. As implied by the colours in Figure 1, these resources are linked to the SC agent's components: The workflow engine executes processes of an agent's BPM, and thus updates its workflow state. Similarly, the reasoning engine reads the SC agent's BRs and turns them into decisions towards actions for each state. The communication environment allows the exchange of messages within the SC

through an appropriate infrastructure. Lastly, two tools analyse the SC simulation results: The SC performance calculator measures its performance, while the SC disruption detector identifies problematic SC operation.

3.2.2 Rule-based framework and implementation

This section explains the main aspects of the implemented simulation system, with respect to the agents' resources and the analysis tools. A rule-based approach is adopted in order to support the automated explanation of SC operation dynamics. This approach is demonstrated here for the most important simulation procedures. More importantly, this section declaratively defines the execution semantics of the formal SC model, which was presented in Section 3.1. For this purpose we provide abstractions in the form of production rules (Giarratano and Riley, 1998), and the syntax to be used is the following:

```
perform operation:
IF ( holds(Condition1)
    AND ...
    AND holds(ConditionN)
    )
THEN ( enforce(Effect1),
    ...
    enforce(EffectM)
    )
```

The *workflow engine* is used by SC agents to execute their BPMs. Its three main operations involve (1) creating BPM instances, (2) executing BP instances and (3) executing junction instances. The last two will be discussed in this section. The rule for process instance execution is provided below. According to it, a process instance is executed if it has been reached, and its trigger conditions and preconditions hold. Once a process instance starts its execution, three effects take place: its execution completion is scheduled, its actions are scheduled for execution and any entities needed for its execution are assigned to it. The developed workflow engine is based on previous work (Manataki and Chen-Burger, 2009) that has been considerably extended. We have specified rules for the holding of individual preconditions and trigger conditions and we have implemented the execution of a wide range of actions. Note that the assignment of entities to a process instance execution guarantees that the

entities needed for its execution are not used by some other process instance. The assigned entities are released once the process instance execution is completed.

```
execute_process_instance(ProcessInst, TriggCond, Precond, Actions):
IF (  reached(ProcessInst)
      AND trigger_conditions_hold(TriggCond)
      AND preconditions_hold(Precond)
    )
THEN (  schedule_execution_completion(ProcessInst),
        schedule_actions_execution(Actions),
        assign_entities(ProcessInst)
    )
```

The rule for junction instance execution is provided below. According to it, a junction instance is executed if its type conditions hold according to the FBPML specification (Chen-Burger et al. 2002). Once a junction instance is executed, the process instances directly following it are considered to be reached.

```
execute_junction_instance(JunctionInst, PostProcessInsts):
IF junction_type_satisfied(JunctionInst)
THEN reach(PostProcessInsts)
```

The *reasoning engine* is used by SC agents to execute their business rules. It enables the execution of three kinds of BRs: (1) policy- and flexibility-BRs of ifthen(IFpart, THENpart) content form, (2) process precondition BRs and (3) BRs for popular, customised policies, such as the (R,Q) policy. The last two will be discussed in this section. The rule for executing a BR of ifthen/2 content form is provided below. According to it, a BR of this type is executed if its IFpart is satisfied, enforcing the effects specified in THENpart.

```
execute_ifthenBR(BrId, IfPart, ThenPart):
IF br_condition_holds(IfPart) THEN enforce_effects(ThenPart)
```

The rule for executing process precondition BRs prescribes that such a BR is executed if the conditions expressed through its content hold. Note that no effects are enforced with its execution; nevertheless, the execution of such a BR can lead to the execution of the corresponding process instance, as it contributes to the satisfaction of its preconditions.

The *communication environment* allows the agent to read and send messages to other SC members. Sending a message is considered to be a BP action that is executed

through the communication environment. Once the sending of a message is invoked, a message is created and transferred to its recipients. The rule for reading messages is provided below, and it assumes full trust between SC members. According to it, a message is read if it is received by the agent. The effect of reading an inform-message is that its content is added to the SC agent's knowledge base in the form of a fact.

```
read_message(MessageId, Content):  
IF received(MessageId) THEN create_fact(Content)
```

The *SC performance calculator* reads the simulation results for SC operation and computes the supply chain performance. We have implemented the calculation for following performance metrics: individual SC members' cost, on time rate and cycle times, as well as total SC cost. The formulae for calculating these metrics follow the corresponding specification of the SCOR model.

The *SC disruption detector* identifies problematic SC operation, as described in Section 3.1. The process of detecting certain types of problematic situations is simple, while for others it is more complex. For example, the cancellations of order deliveries are typically communicated between SC members through messages, and thus they are tracked through the filtering of message content. Start and finish delays of process instances are detected by comparing the actual to the expected execution time start or completion, respectively. Low SC performance is detected by comparing its actual to its expected or desired value. We regard the detection of SC disruptions as an advantage of this approach compared to existing work in SC simulation.

3.2.3 Simulation algorithm

So far we have presented the rule-based operations of different simulation modules. But when are these performed and how are they combined in order to simulate SC operation? The top-level simulation algorithm, which is represented in Figure 2 in the form of an activity diagram, answers precisely this question. Two parts can be seen: a cyclic simulation part, which shows the sequence of simulation steps for each timepoint, and a part that corresponds to the steps at the end of simulation. Note that the user specifies the time period for which he/she wants to run the simulation (e.g. 23 timepoints), and hence simulation ends once this timepoint is reached. The coloured steps in the diagram have already been discussed, and the colour of each step corresponds to the module in which it takes place (as in Figure 1). The white steps involve simulation aspects at the top level, such as initialising

simulation based on the simulation input, and updating the time at the end of each simulation cycle. The ‘Enforce modifications’ step enforces any modifications relevant to the current timepoint, such as occurred errors with items and lot size changes. The names of several steps in Figure 2 end with ‘for all’. This means executing the step for all SC agents, one after the other.

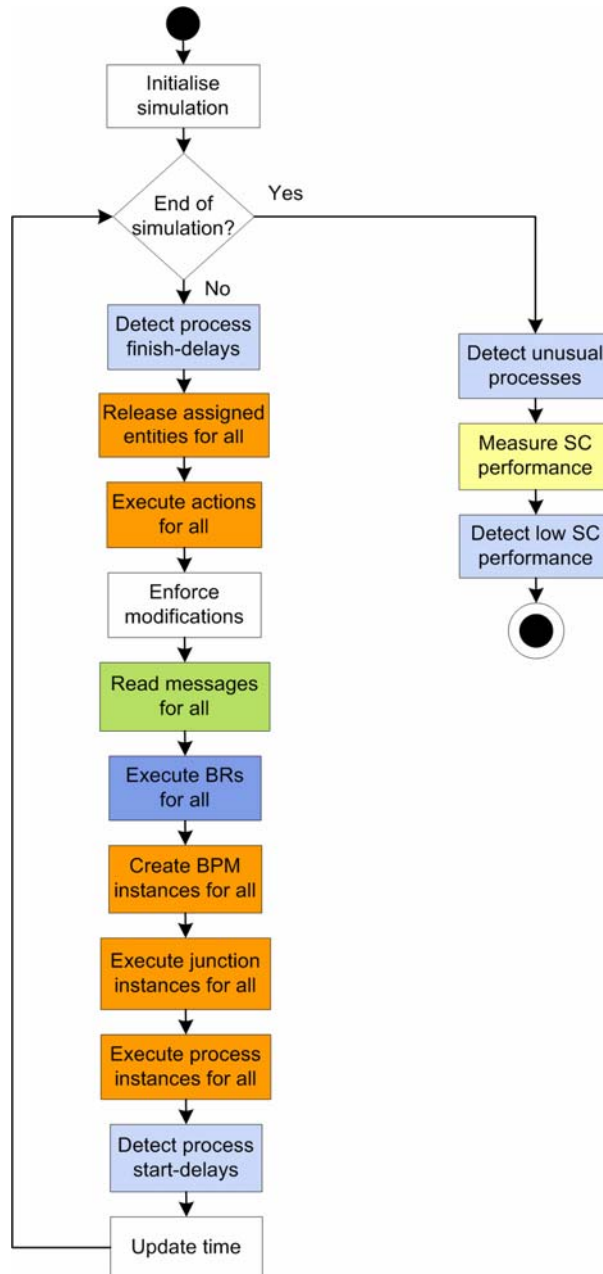


Fig. 2: Simulation algorithm

3.3 Explaining SC operation

This section presents a mechanism for generating detailed explanations of simulation results. This way, a deep understanding of interdependencies across the SC can be gained, thus addressing the first limitation of existing work identified in Section 2.

3.3.1 Knowledge-based framework

The analysis of SC operation dynamics involves explaining the simulation results with respect to four topics: (1) SC operational behaviour, (2) the state of the SC at a certain timepoint, (3) SC performance and (4) detected problematic SC operation. We believe that the most important type of question to ask on these topics is “why”. For example, one might want to find out why a particular process instance is executed at some SC member at some timepoint or why a specific product is available at some SC member at some timepoint. Similarly, the user might be interested to know how the on time rate for some SC member was calculated and why a finish delay was detected for a particular process instance.

Answering such questions is based on the rule-based execution semantics of the formal SC operation model, the choice of which facilitates the explanation process. Figure 3 shows how SC operation can be explained given the production rule-based notation for describing execution semantics that was introduced in Section 3.2.2. According to it, an operation is performed because all its conditions hold, and some $Effect_i$ is enforced because the operation is performed. Let us clarify that for some $Condition_j$ to hold, the current SC state needs to be appropriate, as shaped by the enforcement of performed operations’ effects.

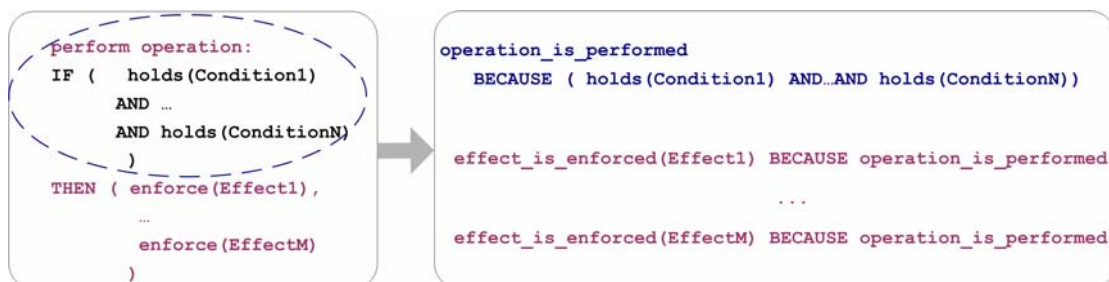


Fig. 3: Explaining execution semantics

3.3.2 Implementation

The explanation of simulation results is implemented based on the formal execution semantics discussed in Section 3.2.2 and the mapping illustrated in Figure

3. The main idea involves keeping a simulation log that contains causal information, and deriving explanations based on this causal information. These two matters are further described in this section.

The simulation log is a report of interesting simulation events (here, by ‘events’ we do not refer to SC events but to incidents that take place during simulation), such as the execution of process instances and the reading of messages by some SC member. This report does not only contain information on the simulation events that take place, but also on the reasons for which these take place. These reasons are deduced based on the formal execution semantics, as translated in Figure 3.

In our implementation, the simulation log contains information of the form `fact(SimulationEvent, ListOfReasons, Timepoint)`. Three illustrative examples follow. According to the first fact, the entity `r-man-462` of type `Product5` is moved at timepoint 22 from `Manufacturer` to `Retailer2`; this happens because the action of moving such an entity is a post-condition of process instance `bpm-515/man_d112`, which finishes its execution at timepoint 22. According to the second fact, the `Manufacturer`’s on time rate is found to be 0.88 at timepoint 38 because `Manufacturer` delivers 17 orders in total, of which 15 are delivered on time. According to the third fact, a finish-delay is detected for process instance `bpm-35/sup1_m16` because its execution is completed at timepoint 9 and not at timepoint 8, as scheduled.

```
fact(entity_is_moved(r-man-462,product5,retailer2,manufacturer),
     [post_condition(move_entity([r-man-462],product5,1,retailer2),
                    bpm-515/man_d112),
     process_finishes_execution(bpm-515/man_d112,22)], 22).
fact(on_time_rate(manufacturer,0.88),
     [number_of_delivered_orders(manufacturer,17),
     number_of_orders_delivered_on_time(manufacturer,15)], 38).
fact(finish_delay_is_tracked(bpm-35/sup1_m16,supplier1,make,m16),
     [process_schedule_finish_time(bpm-35/sup1_m16,8),
     process_actual_finish_time(bpm-35/sup1_m16,9)], 9).
```

Deriving explanations based on a simulation log that contains such causal information is a straightforward task. The process of explaining a simulation event `SimulationEvent`, for which there is relevant information of the form `fact(SimulationEvent, ListOfReasons, Timepoint)` in the simulation log, consists of retrieving its `ListOfReasons`.

It is interesting to note that each derived reason for some simulation event can be further explained following the same explanation process, thus generating a new set of reasons, which can in turn be explained, and so forth. This means that a full explanation tree can be produced, if needed. We have implemented the explanation process in SICStus Prolog (Intelligent Systems Laboratory, 2003), which allows for use of recursion with ease, and facilitates the generation of such an explanation tree.

4. Illustrating example

We have employed the devised knowledge-based framework and implemented systems for the analysis of the operation dynamics of a hypothesised SC scenario. The supply chain studied is realistic, consisting of several SC members arranged in a non-linear structure. It is also a rich scenario, in which SC members exhibit flexible behaviours as a reaction to the occurrence of disruptive events. We should emphasise that the investigated SC scenario involves complex operation dynamics, which are successfully analysed with the use of SColog. This section presents this successful use case for SColog, thus demonstrating its contribution. At the same time, it clarifies how the proposed framework for modelling, simulating and explaining SC operation dynamics can be applied to a specific and representative case.

4.1 Example of SC operation modelling

Consider the supply chain that is presented in Figure 4, and which consists of eight members across four tiers. SC members provide one or more types of products to their customers, as shown in Table 1. Note that Supplier5 acts as a backup supplier for Supplier4, accommodating urgent orders very quickly but costly.

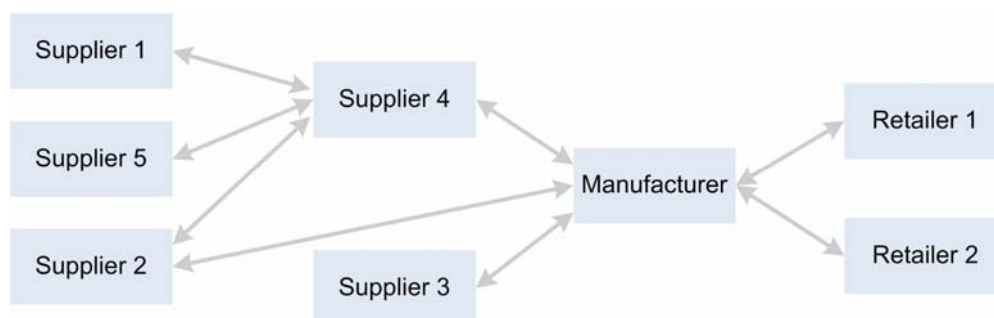


Fig. 4: Example SC structure

SC member	Product	SC operations
Supplier1	Product1 (P1)	make, deliver
Supplier2	Product2 (P2)	make, deliver
Supplier3	Product3 (P3)	deliver
Supplier4	Product4 (P4)	source P1, source P2, make, deliver
Supplier5	Product1 (P1), Product2 (P2)	deliver
Manufacturer	Product5 (P5)	source P4, source P2, source P3, make, deliver
Retailer1		source P5
Retailer2		source P5

Table 1: Products and SC operations for each SC member

All SC members keep inventory of needed products and use resources. They also keep information on different subjects. Examples of formalised structural constructs for Supplier4 are provided below, covering products, inventory, resources and data.

```
supply_chain_member(supplier4).
entity_occ(supplier4, machine, r_sup4_1).
entity_occ(supplier4, product4, r_sup4_2).
entity_occ(supplier4, product4, r_sup4_3).
inventory(supplier4, on_hand, product4, 2, [r_sup4_2,r_sup4_3]).
data(supplier4, product4_production_lot_size, 6).
```

Policies and flexibility BRs drive SC members' decision making. A production policy for Supplier4 is provided below, based on which there is a need for production every 3 timepoints. A flexibility BR for Supplier4 follows, according to which there is a need for urgent sourcing whenever there is a quality issue with P2 on-hand items.

```
br(supplier4, br_sup4_3, policy, ifthen(
  current_time_form_of(3*k), [create_event(need_for_production)])).
br(supplier4, br_sup4_urg1, flexibility_br, ifthen(
  error_with_items(product2, EntityAmount, on_hand),
  [create_event(need_for_product2_urgent_sourcing),
  update_lot_size_if_needed(product2_urglot_size, EntityAmount)])).
```

The SC members' acting behaviour involves sourcing, making and delivering products. Figure 5 presents Supplier4's SCOR-based BPM for making Product4, followed by the declarative specification of the junction and business process that are marked in red. Note that process sup4_m11, which schedules the production for P4, is triggered by an event of type need_for_production (which occurs due to the execution of production policy br_sup4_3 that was presented earlier).

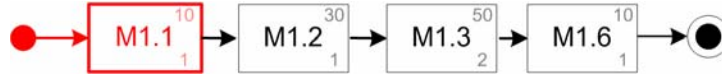


Fig. 5: Supplier4's making BPM

```

junction(supplier4, sup4_jm0, start, [], [sup4_m11]).
process(supplier4, sup4_m11, schedule_product4_production,
  [exist(event_occ(need_for_production))],
  [exist(data(product4_production_lot_size, Product4Amount))],
  [schedule_production(ProdId, product4, Product4Amount),
  create_assigned_event(internal, scheduled_production, ProdId)],
  1, 10).

```

The proposed modelling framework captures the complex operation dynamics of this supply chain, as it considers SC members' behavioural interdependencies with respect to product flow. Furthermore, it expresses the scenario's richness, including e.g. flexibility aspects and urgent sourcing.

4.2 Example of simulating SC operation

The example supply chain that was introduced in the previous section has been simulated for 38 timepoints with the use of the developed simulation system. The numerical values used for simulation are provided in the Appendix. In this section we will present the simulation results, thus illustrating the simulation approach discussed in Section 3.2.

During simulation the user is provided with information on real-time SC operation. This includes information on process instances finishing execution and on the execution of their actions, on the receipt of messages, on the execution of policy- and flexibility-BRs, as well as on the execution of process instances. Through this information the user can have an insight into not only the operational behaviour of SC members but also the flow of products and information involved. Figure 6 presents the output for SC operation at timepoint 4. Note that the output for timepoint 4 is fairly short, while the output for later timepoints (e.g. 28) is considerably longer.

xplainSC T=4

T=1	T=2	T=3	T=4	T=5	T=6	T=7	T=8	T=9	T=10	T=11	T=12	T=13
T=21	T=22	T=23	T=24	T=25	T=26	T=27	T=28	T=29	T=30	T=31	T=32	T=33

Processes finish execution

bpm-7/sup2_m11

Actions

Production 8 for product2 is scheduled by supplier2 as a postcondition of bpm-7/sup2_m11

Event 9 of type scheduled_production occurs by and for supplier2 within BPMinst bpm-7 as a postcondition of bpm-7/sup2_m11

Errors & Messages

The new message 29 is now read by supplier2, and therefore the fact order(27,manufacturer,supplier2,manufacturer,product2,8,10) has now been added to his knowledge base

Policies and Flexibility BRs

Policy br_sup2_1 is now fired within supplier2 causing the following: event 10 of type need_for_production is created

Policy br_man_2 is now fired within manufacturer causing the following: event 11 of type need_for_product2_sourcing is created

Policy br_man_3 is now fired within manufacturer causing the following: event 12 of type need_for_product3_sourcing is created

Processes start execution

Process bpm-7/sup2_m13 starts now execution by supplier2 till timepoint 7

Process bpm-10/sup2_m11 starts now execution by supplier2 till timepoint 5

Process bpm-12/man_s11_p3 starts now execution by manufacturer till timepoint 5

Process bpm-11/man_s11_p2 starts now execution by manufacturer till timepoint 5

Fig. 6: Simulation output for timepoint 4

When simulation is completed, SC performance results are provided. This includes costs, on time rates and cycle times for each SC member, as well as total SC cost. Furthermore, problematic SC operation is detected, and the user is informed about low SC performance and SC disruptions, as depicted in Figure 7. For instance, during the operation of the example SC, an error with P2 items occurs at Supplier4, and Supplier1 cancels the delivery of two orders.

Process duration delay

Process bpm-35/sup1_m16 that starts execution by agent supplier1 at timepoint 7 has a duration delay of 1

Process bpm-110/trans2_d112 that starts now execution by agent transporter2 at timepoint 8 has a duration delay of 1

Process bpm-362/sup1_d13 that starts execution by agent supplier1 at timepoint 15 has duration delay of 2

Error with items

An error occurs at supplier1 at timepoint 10: 6 items of type product1 and status reserved/123 are no longer available

An error occurs at supplier4 at timepoint 16: 4 items of type product2 and status on_hand are no longer available

An error occurs at supplier1 at timepoint 31: 6 items of type product1 and status reserved/1162 are no longer available

Big order

Order 274 for 24 product2 placed by agent supplier4 is received by agent supplier2 at timepoint 13 through bpm-275/sup2_d12 and it is found to be bigger than usually

Order 1170 for 6 product4 placed by agent manufacturer is received by agent supplier4 at timepoint 30 through bpm-1171/sup4_d12 and it is found to be bigger than usually

Cancellation of order delivery

Supplier1 cancels the delivery of order 123 at timepoint 10

Supplier1 cancels the delivery of order 1162 at timepoint 31

Fig. 7: Extract of detected SC disruptions

The simulation of the example supply chain illustrates how this work covers the last two of the three identified limitations of existing SC simulation approaches. Firstly, SC disruptions are detected and the user is accordingly notified. Secondly, decision-making for flexibility purposes is simulated, thus covering SC agility aspects. Explaining the simulation results and analysing the operation dynamics of this scenario is not an easy task for SC managers, given the complex interrelationships

between SC members and their operational behaviours. This task can be undertaken by the developed explanation system, thus addressing the first limitation in SC simulation literature. This is further discussed in the following section.

4.3 Example of explaining SC operation

The simulation results for the example supply chain can be explained with the use of the explanation system that was presented in Section 3.3. In this section we will present four examples of generated explanations with respect to the four points of the research problem, thus demonstrating the value of automated explanation support. Additional examples of generated explanations can be found in Manataki (2012).

The first example involves identifying interrelationships between different aspects of an SC member's operational behaviour. The explanation system discovers that the sourcing BR that fired the execution of process instance bpm-138/man_s11_p2 for the sourcing of 8 P2 items was business rule br_man_2b. This information is useful for understanding the reasons behind this sourcing activity. Discovering such information without automated explanation support would be a challenging task, given that Manufacturer has several alternative sourcing policies.

The second example involves discovering interrelationships between the operational behaviour of different SC members. More specifically, the explanation system identifies that the cancellation of an order delivery by Supplier1 leads to the execution of a flexibility BR at Supplier4. Figure 8 presents an extract of the generated explanation tree. It can be seen that Supplier4's reasoning based on BR br_sup4_urg2 is a reaction for agility purposes to Supplier1's cancellation.

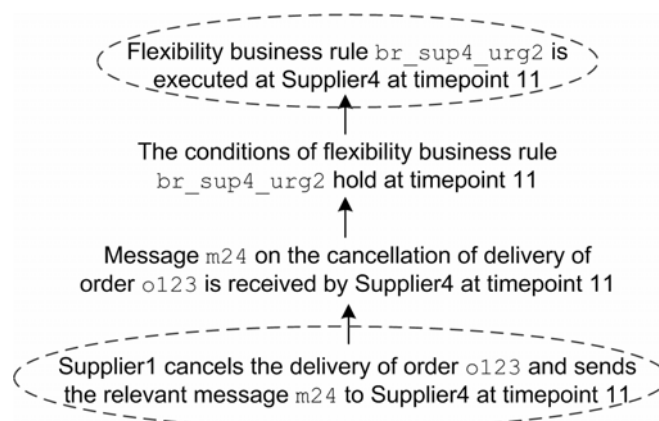


Fig. 8: Explanation tree that illustrates interrelationships between the operational behaviours of Supplier4 and Supplier1

The third example involves discovering interrelationships between SC members' operational behaviour and SC performance. For instance, the explanation of measured cost for an SC member includes information on the process instances that were executed by this SC member as well as on their individual costs. This way the acting behaviour of SC members is directly linked to their performance. This information is particularly useful in the case of low SC performance, where one can trace its causes with the use of the provided explanation.

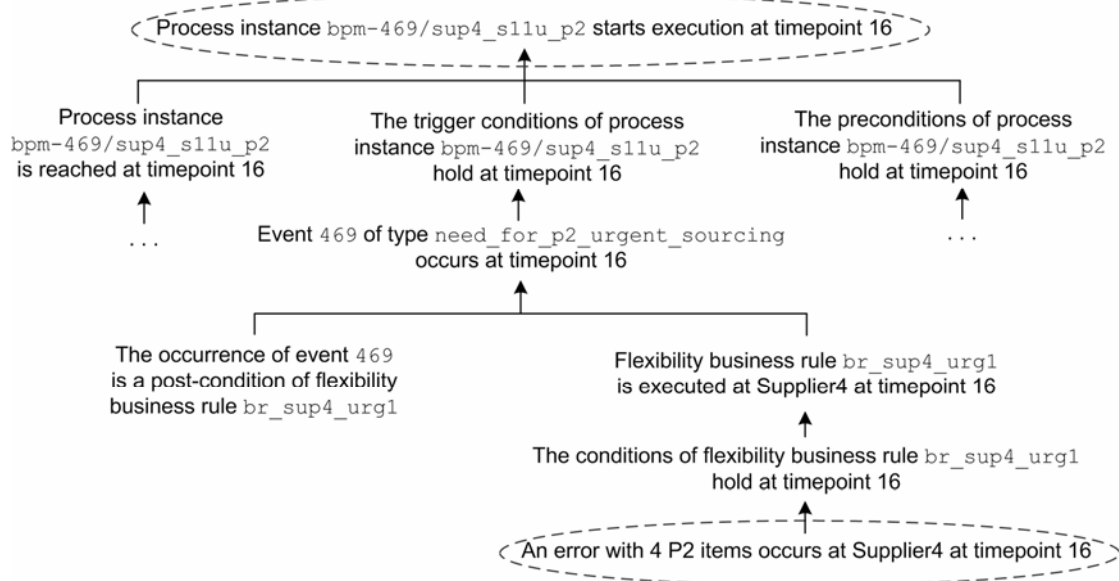


Fig. 9: Explanation tree that illustrates interrelationships between an SC disruption and the operational behaviour of Supplier4

The fourth example involves the identification of interrelationships between SC members' operational behaviour and SC disruptions. Several disruptive events occur at different SC members during simulation, and diagnosing their causes or effects is challenging for SC managers. Explanation trees that are generated with the use of the explanation system facilitate this task. Such an explanation tree is provided in Figure 9, based on which one can conclude that the error with P2 items that occurs at Supplier4 at timepoint 16 gives rise to urgent sourcing for P2.

These examples demonstrate that SColog can explicitly and rigorously analyse the operation dynamics of a complex SC scenario. The interrelationships discovered with the use of the explanation system would have been particularly hard to derive manually based on the simulation results of Section 4.2. More importantly, the explanation system provides an insight on causes and effects of SC operational behaviours, thus helping identify underlying reasons behind uncommon or

problematic situations; the second and fourth examples illustrate this fact. The explanations provided with the use of SCOLog are valuable, as they can support the coordination of SC activities within a single organisation or across multiple SC members. We believe that they can also guide organised efforts towards SC improvement, as reasons for low SC performance or SC disruptions can be automatically discovered.

5. Evaluation

This section presents the evaluation framework and results with respect to the two research claims stated in Section 1. The first claim is empirically evaluated through appropriately designed experiments. The second claim is analytically evaluated and illustrating examples are discussed.

5.1 Improvement of understanding

An experiment was designed to test the performance of participants in two similar settings, and study any performance improvement involved. In order to guarantee the similarity of the two settings, similar questions were asked on similar types of issues in similar scenarios. Two SC scenarios were used for this experiment, Scenario1 and Scenario2, which involved the operation of the example supply chain presented in Section 4. The operation dynamics complexity of the two scenarios was of the same scale. Three questions were asked for each scenario, covering the direct and indirect causes and effects of SC operation aspects. Each question for Scenario1 was similar to a question for Scenario2, focusing on similar issues of SC operation dynamics.

Subjects were asked to answer scenario questions with or without automated explanation support. For each subject, two variables were measured for the answering of each scenario question: the time to answer and the correctness of the provided answer. Time was measured in seconds, and the maximum time available for each question was 360 seconds. The correctness of each answer was graded from 0 to 10 based on an appropriate marking scheme.

	GroupA	GroupB
Scenario1	Y	N
Scenario2	N	N

Table 2: Availability of automated explanation support per group and scenario

The number of subjects in this experiment was 10, all of which were business experts without SCM expertise. These 10 subjects were split in two groups of equal sizes. All subjects answered all questions for the two scenarios, some with and some without automated explanation support. The availability of automated explanation support per group and scenario is visualised in Table 2.

The relative improvement of performance (with respect to correctness of answers and time to provide an answer) was calculated for each subject and pair of questions, which was then averaged over the three pairs of questions. This way, two metrics were available for each participant's performance improvement: correctness and time-efficiency improvement. Table 3 contains relevant results.

Correctness improvement		Time-efficiency improvement	
GroupA	GroupB	GroupA	GroupB
3.82	0.11	0.67	7
3.66	-0.17	7.59	35.01
3.94	0.11	4.76	0
2.95	0.33	3.08	46
0.32	1.44	1.80	2.67
Average = 2.94	Average = 0.367	Average = 3.579	Average = 18.136

Table 3: Relative improvement of correctness and time-efficiency when automated explanation support was previously used (i.e. GroupA) and when not (i.e. GroupB)

5.1.1 Test of correctness improvement

Statistical hypothesis testing was conducted to evaluate the research claim on improvement of correctness, and more specifically the claim that the improvement of correctness when automated explanation support was previously used is bigger than in the case where it was not. The null and alternative hypotheses follow:

- H0: There is no difference in the improvement of correctness of explanations of SC operation dynamics when the explanation system was previously used and when it was not.
- H1: The improvement of correctness of explanations of SC operation dynamics when the explanation system was previously used is bigger compared to the case where it was not previously used.

A one-tailed two sample independent t-test was performed to determine the t value and its corresponding p value in order to accept or reject the null hypothesis. Using the corresponding data of Table 3, the calculated t-value was found to be 3.509.

This value corresponds to a significance level of $p=0.00856$, which is much smaller than the significance level of 0.05. This means that the null hypothesis can be rejected. Hence we can conclude that the improvement of correctness of explanations on SC operation dynamics when the explanation system was previously used is significantly bigger compared to the case where it was not previously used.

5.1.2 Test of efficiency improvement

According to Table 3, the average time-efficiency improvement of GroupA is smaller than the average time-efficiency improvement of GroupB. For this reason, we statistically tested whether the improvement of time-efficiency when automated explanation support was previously used is smaller compared to the case where it was not previously used. The null and alternative hypotheses follow:

- H0: There is no difference in the improvement of time-efficiency for providing explanations of SC operation dynamics when the explanation system was previously used and when it was not.
- H1: The improvement of time-efficiency for providing explanations of SC operation dynamics when the explanation system was previously used is smaller compared to the case where it was not previously used.

Similarly to the test presented in the previous section, a one-tailed two sample independent t-test was performed. The t-value was found to be 1.541, corresponding to a significance level of $p=0.099$, which is higher than the significance level of 0.05. This means that the null hypothesis cannot be rejected. Therefore we cannot conclude that the improvement of efficiency when the explanation system was previously used is smaller compared to the case where it was not previously used.

It should be emphasised that there was a positive improvement of efficiency when the explanation system was previously used. Since this was the case for all members of GroupA, as shown in Table 3, we can conclude that the use of the explanation system improves the efficiency of non-SCM experts.

5.1.3 Discussion

Based on the data collected and the statistical tests performed, two conclusions were drawn: Firstly, explaining SC operation dynamics with the use of the explanation system improves the understanding of the domain for non-SCM experts,

with respect to their efficiency and correctness when providing relevant explanations. Secondly, the improvement of correctness when the explanation system was previously used is significantly higher compared to the case where it was not previously used. Given these two points, one can conclude that the higher degree of correctness improvement achieved through the prior use of the explanation system does not come at the expense of time-efficiency. On the contrary, there is a parallel efficiency improvement. Hence, the first research claim is satisfied.

5.2 Maintainability and reusability

The second research claim is analytically evaluated in this section, thus showing that the proposed knowledge-based approach for modelling, simulating and explaining SC operation allows for maintainability and reusability. Maintainability in software engineering is defined as “the ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment” (IEEE Std. 610.12, 1990). Software reuse is the use of existing software artefacts to develop a new software system (Krueger, 1992).

The SC operation input models, the developed simulation system and the developed explanation system have some properties that contribute towards maintainability and reusability. Specifically, the SC operation models that are specified following the modelling framework described in Section 3.1 are formal, declarative, generic and loosely-coupled. The simulation system that was implemented following the rule-based approach discussed in Section 3.2 is modular, loosely-coupled, cohesive and generic. The explanation system that was developed based on the knowledge-based approach presented in Section 3.3 is generic and declarative. Software maintainability is supported by modularity, loose coupling and high cohesion (Yourdon and Constantine, 1979). Software reusability is facilitated through formal and generic models and procedures (Prieto- Díaz, 1993).

5.2.1 Input model’s maintainability and reusability

An example is provided here to illustrate the input model’s maintainability. Suppose that the model of the example SC of Section 4.1 needs to accommodate the following two changes: Firstly, Manufacturer no longer sources P2 items from Supplier2; Supplier5 is instead his new P2 supplier. Secondly, Supplier4 performs

additional decision-making for flexibility purposes, which triggers urgent production when a complex set of conditions holds, of the form $((A \text{ or } B) \text{ and } (C \text{ or } D)) \text{ or } E$.

Modifying the simulation input for the first change involves simply updating the corresponding data/3 information without modifying any additional sourcing behaviour elements: $\text{data}(\text{manufacturer}, \text{p2_supplier}, \text{supplier2})$ is updated to $\text{data}(\text{manufacturer}, \text{p2_supplier}, \text{supplier5})$. The second change is addressed by adding an appropriate flexibility BR, the IFpart of which has the form $((A \text{ or } B) \text{ and } (C \text{ or } D)) \text{ or } E$. We should emphasise that these modifications neither affect the existing model nor require additional system implementation. This is a considerable advantage.

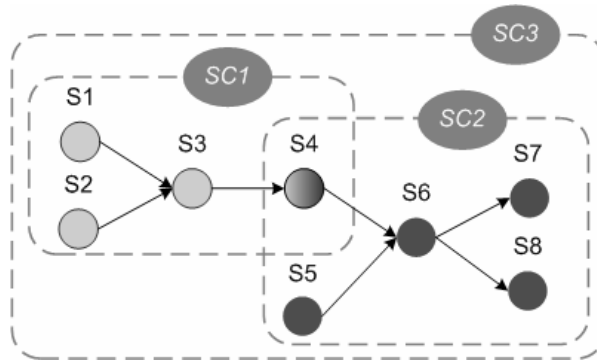


Fig. 10: Merging supply chains SC1 and SC2 results into SC3. In order to specify the input model for SC3 we can reuse the input models for SC1 and SC2.

An example of reusability follows. Consider supply chains SC1 and SC2 presented in Figure 10, and let's suppose that the simulation input for each of these has already been specified (i.e. Input1 and Input2 respectively). If we decide to merge these two supply chains into SC3, then the input model for SC3 fully reuses Input1 and Input2. In fact, no additional information needs to be specified for SC3 input model, i.e. $\text{Input3} = \text{Input1} \cup \text{Input2}$.

5.2.2 Simulation system's maintainability and reusability

Let's suppose that the following two changes require the modification of the simulation system: Firstly, a new version of the SCOR model includes additional processes. Secondly, the assumption of full trust between SC members is relaxed, i.e. the content of received messages is considered only if the message sender is trusted.

No further implementation is required for the first change, given that the additional processes can be represented based on SCORlog's modelling framework. The trust requirement can be encompassed by modifying the communication environment: the related definition of execution semantics is updated as shown below.

Note that this modification does not affect the implementation of any other system component, and it does not require changing the definition of the constructs involved.

```
read_message(MessageId, SenderId, Content):  
IF ( received(MessageId)  
    AND trustworthy(SenderId)  
    )  
THEN create_fact(Content)
```

As far as reusability is concerned, several system components can be used for different applications. For example, the workflow engine is generic enough to be used within a health informatics application for simulating clinical activities.

5.2.3 Explanation system's maintainability and reusability

Two examples are provided to illustrate the explanation system's maintainability. The first example involves the two changes discussed in the previous section. The explanation system does not need to be modified to accommodate these changes, given that appropriate causal information is added in the simulation log. This is also the case for any new or modified simulation models.

The second example aims to show that maintaining the explanation system to answer new types of questions does not require much additional implementation. Let's suppose that we want to identify common reasons for two situations. The implementation of `common_reason(+A, +B, ?C)` in Prolog is provided below. Building on the definition of `reason/2`, only three new lines of code are needed. It is also interesting to note the flexibility provided by this implementation: By giving the goal `common_reason(a, b, C)` we can obtain any common reason for two situations `a` and `b`.

```
common_reason(A, B, C):-  
    reason(A, C),  
    reason(B, C).
```

An example of reusability follows. Let's suppose that we want to explain behaviours for an emergency response scenario, simulated through a different simulation system, which keeps a simulation log with causal information of the form `fact/3`. Given this generic representation, we can use the current explanation system (without any modifications) to answer questions on the emergency response scenario.

6. Conclusions

The research presented in this paper tackled the problem of analysing supply chain operation dynamics, a problem that has been understudied by existing literature. Yet, understanding SC-wide operation dynamics is highly important for coordinating and integrating global supply chains. A knowledge-based solution to this problem was proposed, named SCOLog. SC operation was modelled in a declarative fashion and it was simulated following rule-based execution semantics. This approach facilitated the automated explanation of simulated SC operational behaviours and performance, and it allowed for diagnosing problematic SC operation.

The automated explanation support provided by SCOLog was found to improve the understanding of the domain for non-SCM experts; this has great potential for use in business and SCM education. The maintainability of the approach is a prominent attribute, given the dynamic aspects of SC operation and the evolving nature of the SCM field. Furthermore, SCOLog's reusability makes it possible to explore its potential in a different domain setting. Finally, it is worth mentioning that this work is tailored to the SCM domain, a quality that is beneficial from a practical point of view.

This work has both theoretical and practical implications. As far as theory is concerned, this is the first attempt for an explicit and thorough solution to the problem of analysing SC operation dynamics. We regard this as a contribution to the SCM state of the art, and we hope that it will help begin a fruitful discussion among scholars. At the same time, the research presented in this paper demonstrates the applicability and promising prospects of artificial intelligence techniques for SCM problems, even for issues that haven't been previously explored. It is now time to get the two communities together and benefit from the synergies that can be produced. As far as practical implications are concerned, SCOLog can be readily used by supply chain managers to aid relevant decision-making. The provided simulation and explanation systems can be used to support daily SC operation or SC coordination initiatives. The automated diagnosis of problematic situations can serve as a foundation for SC improvement projects, and the rigorous insight provided can be of great value during SC integration efforts.

Two research limitations must be pointed out. Firstly, stochastic aspects of SC operation are not considered. Nevertheless, only minor modification of SCOLog would be required to simulate and explain SC operation models that include probabilities. Secondly, SCOLog provides an approach to the analysis of operation dynamics of

generic supply chains. The advantage of this design decision is the generality of the solution and the corresponding wide audience. The price to be paid is that some specific requirements of particular business sectors may not be satisfied. Extending SColog to address such issues would be an interesting topic to explore.

A number of avenues of future research are identified, given the contributions and the limitations of this research. Extending our approach for teaching SCM is regarded as a promising direction of future work, especially since SColog was found to improve non-experts' understanding of SCM. To this end, we plan to improve the usability of the simulation and explanation systems by incorporating animation and rich graphical interface. There is also opportunity to employ SColog for studying industry-specific supply chains, such as food supply chains. In this context, we intend to enrich the proposed modelling framework to address requirements on food SCM, such as the ones discussed by van der Vorst et al. (2009). We believe that the approach presented in this paper can serve as a basis for exploring further SCM problems and for studying alternative domains with complex dynamics.

Appendix: Numerical values for case study

SC member	SC operation	Policy (period, quantity)	SCOR-process	Duration	Cost
Supplier1	make	(3, 6)	M1.1	1	10
			M1.3	1	30
			M1.6	1	10
	deliver		D1.2	1	10
			D1.3	1	20
			D1.11	1	20
D1.12			1	50	
Supplier2	make	(1, 12)	M1.1	1	10
			M1.3	3	30
			M1.6	1	10
	deliver		D1.2	1	10
			D1.3	1	20
			D1.11	1	20
			D1.12	2	50
Supplier3	deliver		D1.2	1	10
			D1.3	1	20
			D1.12	1	100
Supplier4	source P1	(3, 6)	S1.1	1	10
			S1.2	1	20
			S1.3	1	10

	source P2	(2, 12)	S1.4	1	10
			S1.1	1	10
			S1.2	1	20
			S1.3	1	10
	make	(3, 6)	M1.1	1	10
			M1.2	1	30
			M1.3	2	50
			M1.6	1	10
	deliver		D1.2	1	10
			D1.3	1	20
			D1.11	1	20
			D1.12	2	50
	Supplier5	deliver		D1	1
Manufacturer	source P2	(2, 8)	S1.1	1	10
			S1.2	1	10
			S1.4	1	10
	source P3	(3, 18)	S1.1	1	10
			S1.2	1	10
			S1.4	1	10
	source P4	(2, 4)	S1.1	1	10
			S1.2	1	10
			S1.4	1	10
	make	(2, 4)	M1.1	1	10
			M1.2	1	20
			M1.3	2	80
			M1.4	1	30
deliver		M1.6	1	10	
		D1.2	1	10	
		D1.3	1	20	
		D1.12	3	100	
Retailer1	source P5	(2, 3)	S1.1	1	10
			S1.2	1	10
Retailer2	source P5	(2, 1)	S1.1	1	10
			S1.2	1	10

Table 4: Policies and processes for each SC member

SC member	Flexibility Business Rule
Supplier1	IF there is an error with P1 items of amount X reserved for order Y THEN cancel delivery of order Y
Supplier4	IF there is an error with P2 items of amount X THEN there is a need for X P2 urgent sourcing
	IF there is a cancellation of an order delivery for P1 items of amount X THEN there is a need for X P1 urgent sourcing

Table 5: Flexibility business rules for each SC member

Product	Component	Quantity
P5	P2	2
	P3	3
	P4	1
P4	P1	1
	P2	4

Table 6: Bills of materials

References

- Adamides, E. D., Papachristos, G., and Pomonis, N. (2012). Critical realism in supply chain research: Understanding the dynamics of a seasonal goods supply chain. *International Journal of Physical Distribution & Logistics Management*, 42(10):906–930.
- Akkermans, H. (2001). Emergent supply networks: System dynamics simulation of adaptive supply agents. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, Washington, DC, USA. IEEE Computer Society.
- Archibald, G., Karabakal, N., and Karlsson, P. (1999). Supply chain vs. supply chain: using simulation to compete beyond the four walls. In Farrington, P., Nembhard, H., Sturrock, D., and Evans, G., editors, *Proceedings of the 1999 Winter Simulation Conference*, pages 1207–1214.
- Arshinder, A. K. and Deshmukh, S. G. (2008). Supply chain coordination: Perspectives, empirical studies and research directions. *International Journal of Production Economics*, 115(2):316–335.
- Axsäter, S. (2006). *Inventory control*. Springer Science, New York, USA, 2nd edition.
- Bagchi, S., Buckley, S. J., Ettl, M., and Lin, G. Y. (1998). Experience using the IBM supply chain simulator. In Medeiros, D., Watson, E., Carson, J., and Manivannan, M., editors, *Proceedings of the 1998 Winter Simulation Conference*, pages 1387–1394.
- Barratt, M. (2004). Understanding the meaning of collaboration in the supply chain. *Supply Chain Management: An International Journal*, 9(1):30–42.
- Blackhurst, J., Craighead, C. W., Elkins, D., and Handfield, R. B. (2005). An empirically derived agenda of critical research issues for managing supply-chain disruptions. *International Journal of Production Research*, 43(19):4067–4081.
- Bolstorff, P. and Rosenbaum, R. (2012). *Supply chain excellence: A handbook for dramatic improvement using the SCOR model*. AMACOM, New York, USA, 3rd edition.
- Burgess, K., Singh, P., and Koroglu, R. (2006). Supply chain management: A structured literature review and implications for future research. *International Journal of Operations & Production Management*, 26(7):703–729.
- Butner, K. (2010). The smarter supply chain of the future. *Strategy & Leadership*, 38(1):22–31.

- Chen-Burger, Y.-H., Tate, A., and Robertson, D. (2002). Enterprise modelling: A declarative approach for FBPML. In *European Conference of Artificial Intelligence (ECAI), Knowledge Management and Organisational Memories Workshop*.
- Choi, T., Dooley, K., and Rungtusanatham, M. (2001). Supply networks and complex adaptive systems: Control versus emergence. *Journal of Operations Management*, 19(3):351–366.
- Chopra, S. and Meindl, P. (2003). *Supply Chain Management. Strategy, Planning & Operation*. Prentice Hall.
- Christopher, M. (2000). The agile supply chain: Competing in volatile markets. *Industrial Marketing Management*, 29(1):37–44.
- Christopher, M. and Peck, H. (2004). Building the Resilient Supply Chain. *International Journal of Logistics Management*, 15(2):1–14.
- Corbett, C. J. (2001). Stochastic inventory systems in a supply chain with asymmetric information: Cycle stocks, safety stocks, and consignment stock. *Operations Research*, 49(4):487–500.
- Craighead, C. W., Blackhurst, J., Rungtusanatham, M. J., and Handfield, R. B. (2007). The severity of supply chain disruptions: Design characteristics and mitigation capabilities. *Decision Sciences*, 38(1):131–156.
- Dong, J., Ding, H., Ren, C., and Wang, W. (2006). IBM SmartSCOR – A SCOR based supply chain transformation platform through simulation and optimization techniques. In Perrone, L., Wieland, F., Liu, J., Lawson, B., Nicol, D., and Fujimoto, R., editors, *Proceedings of the 2006 Winter Simulation Conference*, pages 650–659.
- Fawcett, S. E. and Magnan, G. M. (2002). The rhetoric and reality of supply chain integration. *International Journal of Physical Distribution & Logistics Management*, 32(5):339–361.
- Fiala, P. (2005). Information sharing in supply chains. *Omega*, 33(5):419–423.
- Fox, M., Barbuceanu, M., and Teigen, R. (2000). Agent-oriented supply-chain management. *International Journal of Flexible Manufacturing Systems*, 12(2):165–188.
- Frohlich, M. T. and Westbrook, R. (2001). Arcs of integration: an international study of supply chain strategies. *Journal of Operations Management*, 19(2):185–200.
- Giarratano, J. C. and Riley, G. (1998). *Expert Systems*. PWS Publishing Co., Boston, MA, USA, 3rd edition.
- Giunipero, L., Hooker, R., Joseph-Matthews, S., Yoon, T., and Brudvig, S. (2008). A decade of SCM literature: past, present and future implications. *Journal of Supply Chain Management*, 44(4):66–86.
- Goutsos, S. and Karacapilidis, N. (2004). Enhanced supply chain management for ebusiness transactions. *International Journal of Production Economics*, 89(2):141–152.

- Gunasekaran, A. and Ngai, E. W. T. (2004). Information systems in supply chain integration and management. *European Journal of Operational Research*, 159(2):269–295.
- Gunasekaran, A., Lai, K.-h., and Cheng, T. C. E. (2008). Responsive supply chain: A competitive strategy in a networked economy. *Omega*, Volume 36(4):549–564.
- Harrison, A. and van Hoek, R. (2008). *Logistics Management and Strategy: Competing through the supply chain*. Prentice Hall, Harlow, Essex, UK, 3rd edition
- Holweg, M., Disney, S., Holmström, J., and Småros, J. (2005). Supply Chain Collaboration:: Making Sense of the Strategy Continuum. *European Management Journal*, 23(2):170–181.
- Holweg, M. and Pil, F. K. (2008). Theoretical perspectives on the coordination of supply chains. *Journal of Operations Management*, 26(3):389–406.
- Hwang, H. B. and Xie, N. (2008). Understanding supply chain dynamics: A chaos perspective. *European Journal of Operational Research*, 184(3):1163–1178.
- IEEE Std., 610.12, (1990). IEEE standard glossary of software engineering terminology. IEEE Computer Society Press, Los Alamitos, CA.
- Intelligent Systems Laboratory, Swedish Institute of Computer Science (2003). SICStus Prolog User's Manual, Release 3.10.1.
- Krueger, C. W. (1992). Software reuse. *ACM Computing Surveys*, 24(2):131–183.
- Lee, H., Padmanabhan, V., and Whang, S. (1997). Information distortion in a supply chain: The bullwhip effect. *Management Science*, 43(4):546–558.
- Lee, H. L., So, K. C., and Tang, C. S. (2000). The value of information sharing in a two-level supply chain. *Management Science*, 46(5):626–643.
- Liu, J., Zhang, S., and Hu, J. (2005). A case study of an inter-enterprise workflow-supported supply chain management system. *Information & Management*, 42(3):441–454.
- LlamaSoft Inc. (2012). Supply Chain Guru.
- Longo, F. and Mirabelli, G. (2008). An advanced supply chain management tool based on modeling and simulation. *Computers & Industrial Engineering*, 54(3):570–588.
- Manataki, A. (2012). *Analysing Supply Chain Operation Dynamics through Logic-Based Modelling and Simulation*. PhD thesis, School of Informatics, The University of Edinburgh.
- Manataki A. and Chen-Burger, Y-H (2009). Analysing Supply Chain Strategies using Knowledge-Based Techniques. In Håkansson, A., Nguyen, N.T., Hartung, R.L., Howlett, R.J. and Jain, L.C., editors, *Agents and Multi-Agent Systems: Technologies and Applications*, pages 697–704. Lecture Notes In Artificial Intelligence, Vol. 5559, Springer-Verlag, Berlin, Heidelberg.
- Melnyk, S. A., Lummus, R. R., Vokurka, R. J., Burns, L. J., and Sandor, J. (2009). Mapping the future of supply chain management: A Delphi study. *International Journal of Production Research*, 47(16):4629–4653.

- Persson, F. (2011). SCOR template – a simulation based dynamic supply chain analysis tool. *International Journal of Production Economics*, 131(1):288–294.
- Persson, F. and Araldi, M. (2009). The development of a dynamic supply chain analysis tool – integration of SCOR and discrete event simulation. *International Journal of Production Economics*, 121(2):574–583.
- Piramuthu, S. (2005). Knowledge-based framework for automated dynamic supply chain configuration. *European Journal of Operational Research*, 165(1):219–230.
- Puigjaner, L. and Laínez, J. M. (2008). Capturing dynamics in integrated supply chain management, *Computers & Chemical Engineering*, 32(11):2582–2605.
- Prieto-Díaz, R. (1993). Status report: Software reusability. *IEEE Software*, 10(3):61–66.
- Rai, A., Patnayakuni, R., and Seth, N. (2006). Firm Performance Impacts of Digitally Enabled Supply Chain Integration Capabilities. *MIS Quarterly*, 30(2):225–246.
- Ren, C., He, M., Wang, Q., Shao, B., and Dong, J. (2010). Driving supply chain transformation through a business process oriented approach. *Service Science*, 2(4):298–314.
- Riddalls, C. E., Bennett, S., and Tipi, N. S. (2000). Modelling the dynamics of supply chains. *International Journal of Systems Science*, 31(8):969–976.
- Sarimveis, H., Patrinos, P., Tarantilis, C. D., and Kiranoudis, C. T. (2008). Dynamic modeling and control of supply chain systems: A review. *Computers & Operations Research*, 35(11):3530–3561.
- Simatupang, T., Wright, A., and Sridharan, R. (2002). The knowledge of coordination for supply chain integration. *Business Process Management Journal*, 8(3):289–308.
- Stadtler, H. (2008). Supply Chain Management – An Overview. In Stadtler, H. and Kilger, C., editors, *Supply Chain Management and Advanced Planning: Concepts, Models, Software and Case Studies*, pages 9–36. Springer-Verlag, Berlin, 3rd edition.
- Stefanovic, D., Stefanovic, N., and Radenkovic, B. (2009). Supply network modelling and simulation methodology. *Simulation Modelling Practice and Theory*, 17(4):743–766.
- Stefik, M. (1995). *Introduction to Knowledge Systems*. Morgan Kaufmann, San Francisco, CA, USA.
- Sterman, J. D. (2006). Operational and Behavioral Causes of Supply Chain Instability. In Carranza, O. and Villegas, F., editors, *The Bullwhip Effect in Supply Chain*, pages 17–56. Palgrave MacMillan, UK.
- Stock, J., Boyer, S., and Harmon, T. (2010). Research opportunities in supply chain management. *Journal of the Academy of Marketing Science*, 38(1):32–41.
- Storey, J., Emberson, C., Godsell, J., and Harrison, A. (2006). Supply chain management: theory, practice and future challenges. *International Journal of Operations & Production Management*, 26(7):754–774.

- Supply Chain Council (2008). Supply Chain Operations Reference Model, Version 9.0.
- Swaminathan, J. M., Smith, S. F., and Sadeh, N. M. (1998). Modeling supply chain dynamics: A multiagent approach. *Decision Sciences*, 29(3):607–632.
- The Business Rules Group (2000). Defining business rules – what are they really?
- van der Aalst, W.M.P. and Stahl, C. (2011) *Modeling Business Processes: A Petri Net Oriented Approach*, MIT Press, Cambridge, MA.
- van der Vorst, J. G., Tromp, S.-O., and van der Zee, D.-J. (2009). Simulation modelling for food supply chain redesign; integrated decision making on product quality, sustainability and logistics. *International Journal of Production Research*, 47(23):6611–6631.
- Yu, Z., Yan, H., and Cheng, T. C. E. (2001). Benefits of information sharing with supply chain partnerships. *Industrial Management & Data Systems*, 101(3):114–121.
- Yourdon, E. and Constantine, L. L. (1979). *Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design*. Prentice Hall, Upper Saddle River, NJ, USA, 1st edition.
- Wooldridge, M. J. (2002). *Introduction to MultiAgent Systems*. John Wiley & Sons, Inc., New York, NY, USA.