

# Loosely Coupled Interorganizational Workflows: modeling and analyzing workflows crossing organizational boundaries

Wil van der Aalst

Department of Mathematics and Computing Science  
Eindhoven University of Technology  
P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands

telephone: +31 40 2474295/2733

telefax: +31 40 2463992

e-mail: [wsinwa@win.tue.nl](mailto:wsinwa@win.tue.nl)

Original Submittal: September 22, 1998

Accepted with modification: July 24, 1999

Section: Research

## ABSTRACT

Today's corporations often must operate across organizational boundaries. Phenomena such as electronic commerce, extended enterprises, and the Internet stimulate cooperation between organizations. Therefore, it is interesting to consider workflows distributed over a number of organizations. Interorganizational workflow offers companies the opportunity to re-shape business processes beyond the boundaries of their own organizations. Two important questions are addressed in this paper: (1) What are the minimal requirements any interorganizational workflow should satisfy?, and (2) How does one decide whether an interorganizational workflow, modeled in terms of Petri nets, is consistent with an interaction structure specified through a message sequence chart?

## Keywords

Workflow management, electronic commerce, verification, Petri nets, message sequence charts

## 1. INTRODUCTION

Recently, many Workflow Management Systems (WFMSs) have become available. The phenomenon workflow management will have a tremendous impact on the next generation of information systems [8]. As the workflow paradigm continues to infiltrate organizations that need to cope with complex administrative processes, the WFMS will become a fundamental building block. At the moment, there are more than 200 workflow products commercially available and many organizations are introducing the technology to support their business processes: It is becoming a mature technology that can be applied *within* organizations. However, the number of business processes where multiple organizations are involved is also increasing rapidly. Technologies such as Electronic Data Interchange (EDI), the Internet, and the World Wide Web (WWW) enable multiple organizations to participate in shared business processes. The rise of Electronic Commerce (EC), virtual organizations, and extended enterprises highlights the fact that more and more business processes are crossing organizational boundaries [9]. This paper focuses on interorganizational workflows, i.e., with several business partners involved in shared processes. We restrict ourselves to structured processes with a predefined set of tasks and routing constructs. In cases where the coordination structure and the interaction between the business partners is not specified explicitly, this is not a realistic assumption [12]. Nevertheless, there are numerous situations where the organizations feel a need to specify the coordination structure explicitly. Interorganizational workflow will bring benefits but also technical problems. Therefore, the Workflow Management Coalition (WfMC [10]) is working on standards to enable *workflow interoperability*. These standards address the technical issues but not the content of the coordination structure. It is our belief that the semantics of the constructs needed to model interorganizational workflows should be defined before solving the technical issues (which are mainly syntactical). Therefore, we focus on modeling and analyzing interorganizational workflows. There are several mechanisms to enable them:

- *capacity sharing*: tasks are executed by external resources under the control of one workflow manager,
- *chained execution*: the process is divided into subsequent phases and each business partner takes care of one phase,
- *subcontracting*: a subprocess is executed by another organization,
- *case transfer*: each partner uses the same workflow process and cases are transferred from one partner to another,
- *loosely coupled*: each partner takes care of a specified part of the process.

This paper focuses on loosely coupled workflow processes. The communication mechanism used for interaction is *asynchronous communication*. Loosely coupled workflow processes operate essentially independently, but have to

synchronize at certain points to ensure the correct execution of the overall business process. Synchronization of parallel processes is known to be a potential source of errors (e.g., deadlock and livelocks). Therefore, it is difficult to establish the validity of complex interorganizational workflows. This paper introduces a notion that should be satisfied by any interorganizational workflow. Based on this, we also present an analysis technique to verify the validity of an interorganizational workflow. Moreover, we show how to check whether the interorganizational workflow is consistent with the communication structure (i.e., the protocol) in terms of message sequence charts.

Because processes are a dominant factor in workflow management, it is important to use an established framework for modeling and analyzing workflow processes. In this paper we use a framework based on *Petri nets* [13]. There are several reasons for using Petri nets for workflow modeling: their formal semantics, graphical nature, expressiveness, analysis techniques and tools [1,6].

## 2. THE CONTROL-FLOW PERSPECTIVE

Before focussing on the problems related to interorganizational workflows, we consider the perspectives that are relevant for workflow modeling and workflow execution: (1) *control-flow* (or process) perspective, (2) *resource* (or organization) perspective, (3) *data* (or information) perspective, (4) *task* (or function) perspective, and (5) *operation* (or application) perspective. (These perspectives are similar to the perspectives given in [8].) In the control-flow perspective, *workflow process definitions* (workflow schemas) are defined to specify which tasks need to be executed and in what order (i.e., the routing or control flow). A *task* is an atomic piece of work. Workflow process definitions are instantiated for specific *cases* (i.e., workflow instances). Examples of cases are a request for a mortgage loan, an insurance claim, a tax declaration, an order, or a request for information. Since a case is an instantiation of a process definition, it corresponds to the execution of concrete work according to the specified routing. In the resource perspective, the organizational structure and the population are specified. The organizational structure describes relations between roles (resource classes based on functional aspects) and groups (resource classes based on organizational aspects). Thus clarifying organizational issues such responsibility, availability, and authorization. *Resources*, ranging from humans to devices, form the organizational population and are allocated to roles and groups. The data perspective deals with *control* and *production data*. Control data are data introduced solely for workflow management purposes, e.g., variables introduced for routing purposes. Production data are information objects (e.g., documents, forms, and tables) whose existence does not depend on workflow management. The task perspective describes the elementary operations performed by resources while executing a task for a specific case. In the operational perspective the elementary actions are described. These actions are often executed using applications ranging from a text editor to custom build applications to perform complex calculations. Typically, these applications create, read, or modify control and production data in the information perspective.

This paper addresses the problem of workflow verification. Although each of perspectives is relevant, we focus on the control-flow perspective. In fact, we focus on the life cycle of one case in isolation.

We abstract from the resource perspective because, given today's workflow technology, at any time there is only one resource working on a task which is being executed for a specific case. In today's workflow management systems it is not possible to specify that several resources are collaborating in executing a task. Note that even if multiple persons are executing one task, e.g., writing a report, only one person is allocated to that task from the perspective of the workflow management system: This is the person that selected the work item from the in-basket (i.e., the electronic worktray).

We also (partly) abstract from the data perspective. We abstract from production data because these are outside the scope of the workflow management system. These data can be changed at any time without notifying the workflow management system. In fact their existence does not even depend upon the workflow application and they may be shared among different workflow processes, e.g., the bill-of-material in manufacturing is shared by production, procurement, sales, and quality control processes. The control data used by the workflow management system to route cases are managed by the workflow management system. However, some of these data are set or updated by humans or applications. For example, a decision is made by a manager based on intuition or a case is classified based on a complex calculation involving production data. Clearly, the behavior of a human or a complex application cannot be modeled completely. Therefore, some abstraction is needed to incorporate the data perspective when verifying a given workflow. The abstraction used in this paper is the following. Since control data (i.e., workflow attributes such as the age of a customer, the department responsible, or the registration date) are only used for the routing of a case, we incorporate the routing decisions but not the actual data. For example, the decision to accept or to reject an insurance claim is taken into account, but not the actual data where this decision is based on. Therefore, we consider each choice to be a non-deterministic one and abstract from the workflow attributes.

For similar reasons we (partly) abstract from the task and operation perspectives. We consider tasks to be atomic and abstract from the execution of operations inside tasks. The workflow management system can only launch applications or trigger

people and monitor the results. It cannot control the actual execution of the task. Therefore, from the viewpoint of verification, it is reasonable to focus on the control-flow perspective. In fact, it suffices to consider the life cycle of one case in isolation. The only way cases interact directly is via the competition for resources and the sharing of production data. (Note that control data are strictly separated.) Therefore, if we abstract from resources and data, it suffices to consider one case in isolation. The competition between cases for resources is only relevant for performance analysis.

### 3. AN EXAMPLE

To illustrate our concepts and techniques, we model a workflow that involves four business partners: a customer, a producer and two suppliers. The customer orders a product by sending an order for product *a* to the producer. To produce the ordered product, the producer orders the products needed for production (*b* and *c*). Then the customer is informed that the order has been accepted. Supplier 1 produces products of type *b*, supplier 2 produces products of type *c*. After both products have been delivered, they are assembled into a product of type *a* which is delivered to the customer. After delivery an invoice is sent which is then paid by the customer.

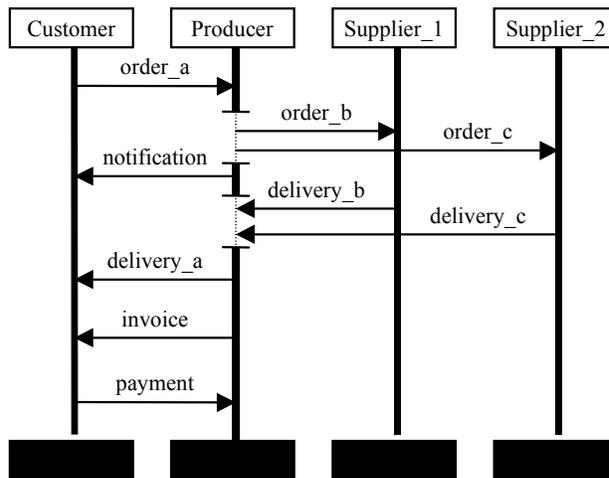
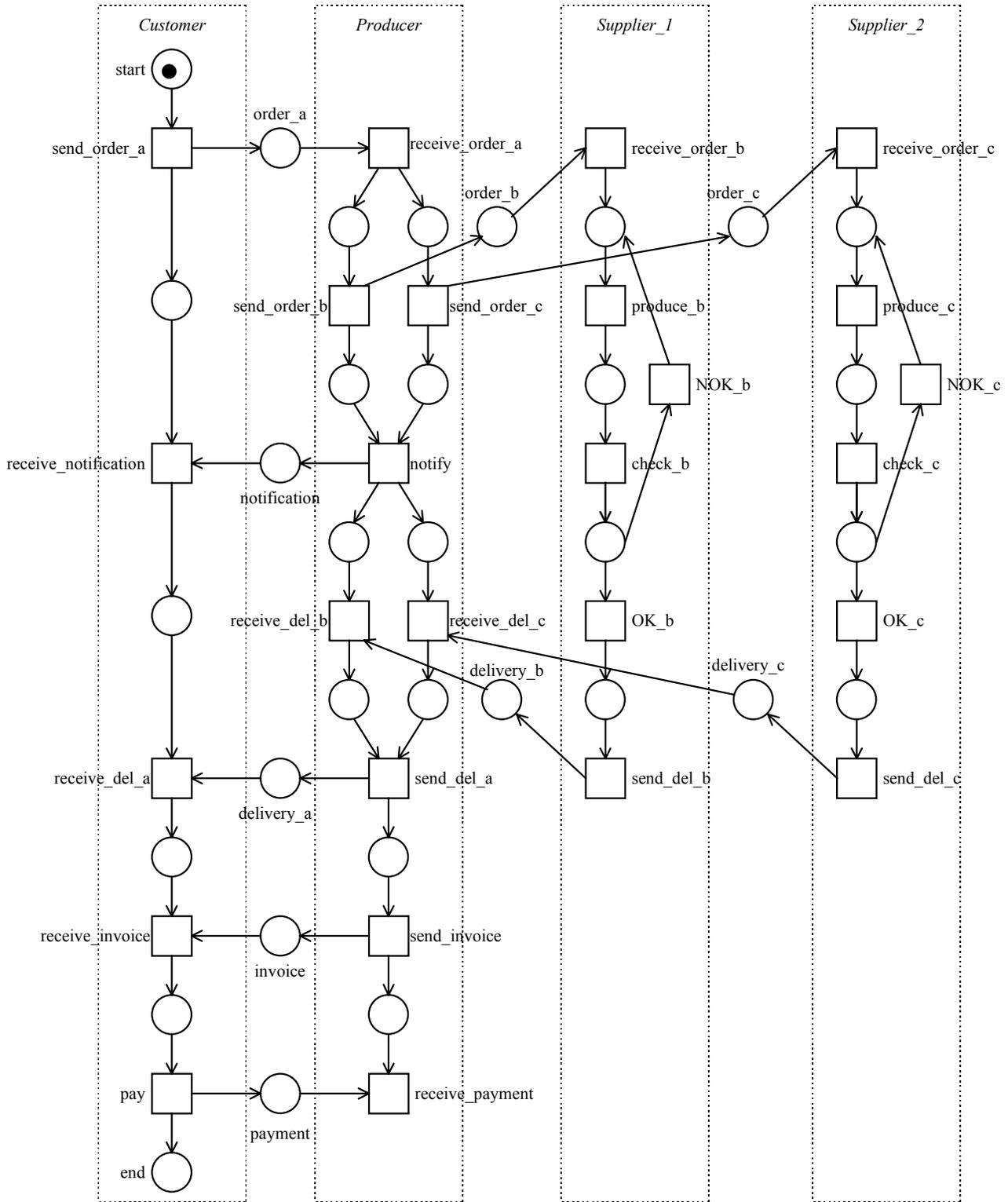


Figure 1: The interaction specified in terms on an MSC.

Figure 1 models the interaction between the four business partners in terms of a *Message Sequence Chart* (MSC). It specifies the messages that are exchanged and the ordering of events associated with sending and receiving messages. In this particular example there are five kinds of messages: orders, notifications, deliveries, invoices, and payments.



**Figure 2: An example of an interorganizational workflow.**

Figure 2 specifies the internal behavior of each business partner in terms of a *Petri net* [12]. It is a network composed of squares and circles. The squares are *transitions* and correspond to tasks that need to be executed. The circles are used to represent the state of a workflow and are called *places*. The arrows between places and transitions specify causal relations. A place  $p$  is called an input place of a transition  $t$  if and only if there exists a directed arc from  $p$  to  $t$ . Place  $p$  is called an output place of transition  $t$  if and only if there exists a directed arc from  $t$  to  $p$ . At any time a place contains zero or more tokens, drawn as black dots. The state of the net, often referred to as *marking*, is the distribution of tokens over places. In Figure 2, only place *start* contains a token. The number of tokens may change during the execution of the net. Transitions are the active components in a Petri net: They change the state of the net according to a firing rule:

1. A transition  $t$  is said to be enabled if and only if each input place  $p$  of  $t$  contains at least one token.
2. An enabled transition may fire. If transition  $t$  fires, then  $t$  consumes one token from each input place  $p$  of  $t$  and produces one token for each output place  $p$  of  $t$ .

By using this rule it is possible to determine which transitions can fire and in what order, i.e., the net specifies the internal behavior of each business partner and the interaction between the partners. Note that the interorganizational workflow shown in Figure 2 is composed of four local workflows.

In this paper we focus on two questions:

- a) Is the interorganizational workflow sound, i.e., does the workflow satisfy some basic properties such as the absence of deadlocks and livelocks?
- b) Is the behavior of the interorganizational workflow consistent with the message sequence chart specifying the desired interaction structure (protocol)?

#### 4. MESSAGE SEQUENCE CHARTS

Interorganizational workflows are described in terms of individual tasks and causal relations. In most cases, the design of an interorganizational workflow starts with the specification of the communication structure. Clearly, a description in terms of a Petri net is too detailed to start with. Therefore, another technique is needed. We use *Message Sequence Charts* (MSCs) for this purpose. MSCs are a graphical language for the visualization of communications between systems/processes [7,14]. The representation of the MSC is intuitive and focuses on the messages between communication entities. The actors are termed *business partners* or *instances*, which communicate via messages such as *order\_a*, *order\_b*, *order\_c*, *delivery\_a*, *delivery\_b*, *delivery\_c*, *notification*, *invoice*, and *payment*. Each message has a sender and a receiver. Within each instance, events are ordered, e.g., the instance *Supplier\_1* sends *delivery\_b* only after the receipt of *order\_b*. The ordering of events is specified by the time axis of an instance which is represented by a vertical line. In an MSC, it is also possible to specify *coregions* which are represented by a dashed part of the time axis of an instance. Events in a coregion are unordered in time. In Figure 1 there are two coregions, e.g., the events sending *order\_b* and sending *order\_c* are in one coregion. We use a variant of MSCs as defined in [11]. The basic chart has been extended with coregions. However, for reasons of simplicity, we do not allow for other customary extensions such as process creation, process termination, timers, and refinement.

#### 5. INTERORGANIZATIONAL WORKFLOWS

Workflows are *case-based*; i.e., every piece of work is executed for a specific case. Examples of cases are a mortgage, an insurance claim, a tax declaration, an order, or a request for information. Cases are often generated by an external customer. However, it is also possible that it is generated by another department within the same organization (internal customer). A workflow process is designed to handle similar cases. Cases are handled by executing tasks in a specific order. The workflow process definition specifies which tasks need to be executed and in what order. Recall that for the purpose of verification we focus on the control-flow perspective and abstract from the other perspectives mentioned in Section 2. In the workflow process definition, building blocks such as the AND-split, AND-join, OR-split and OR-join are used to model sequential, conditional, parallel and iterative routing. Clearly, a Petri net can be used to specify the routing of cases. Tasks are modeled by transitions and causal dependencies are modeled by places. In fact, a place corresponds to a condition which can be used as pre- and/or post-conditions for tasks. An AND-split corresponds to a transition with two or more output places, and an AND-join corresponds to a transition with two or more input places. OR-splits/OR-joins correspond to places with multiple outgoing/ingoing arcs. Moreover, in [1] it is shown that the Petri net approach also allows for useful routing constructs absent in many WFMSs.

A Petri net that models the process aspect of a workflow, is called a *WorkFlow net* (WF-net). It should be noted that it specifies the dynamic behavior of a single case in isolation. A WF-net is a Petri net with one source place (i.e., with no

ingoing arcs) and one sink place (i.e., with no ingoing arcs), and every node is on a path from the source place to the sink place.

A WF-net has one source and one sink because any case handled by the procedure represented by the WF-net is created if it enters the WFMS and is deleted once it is completely handled by the WFMS. Moreover, any node should be on a path from the source to the sink. This requirement has been added to avoid ‘dangling tasks and/or conditions’, i.e., tasks and conditions which do not contribute to a process. Each of the four subprocesses in Figure 2 is an example of a WF-net.

The application of Petri nets to the modeling and analysis of workflows within an organization has been reported in [1,6]. An *interorganizational workflow* is essentially a set of loosely coupled workflow processes. Typically, there are  $n$  business partners which are involved in one ‘global’ workflow process. Each partner has its own ‘local’ workflow process which is private, i.e., the corresponding business partner has full control over the local part of the workflow. However, these local workflow processes need to communicate. The global workflow process therefore consists of local workflow processes and an interaction structure. There is just one way to interact: asynchronous communication by exchanging messages between local workflow processes. Each of the local workflows shown in Figure 2 is described by a WF-net. It should be noted that three of the four local workflows do not have an explicit source and sink. However, this is merely a technical issue. The WF-net corresponding to the producer is activated by an order from the customer and terminates after receiving the payment. Therefore, as Figure 3 shows, an explicit source and sink can be added in a straightforward manner.

## 6. VERIFICATION

Errors in the design of an interorganizational workflow are difficult to trace and may have serious consequences. Moreover, the problems resulting from an incorrect interorganizational workflow are difficult to repair because of the distributed control. We use Figure 3 to illustrate potential errors. Four changes have been made:

1. *Alteration 1*: if the result of the check by *Supplier\_2* is negative, then task *send\_del\_c* is skipped.
2. *Alteration 2*: task *NOK\_b* in *Supplier\_1* has an extra input place.
3. *Alteration 3*: *receive\_del\_b* and *receive\_del\_c* in *Producer* are executed sequentially instead of in parallel.
4. *Alteration 4*: *send\_del\_a* and *send\_invoice* in *Producer* are executed in parallel.

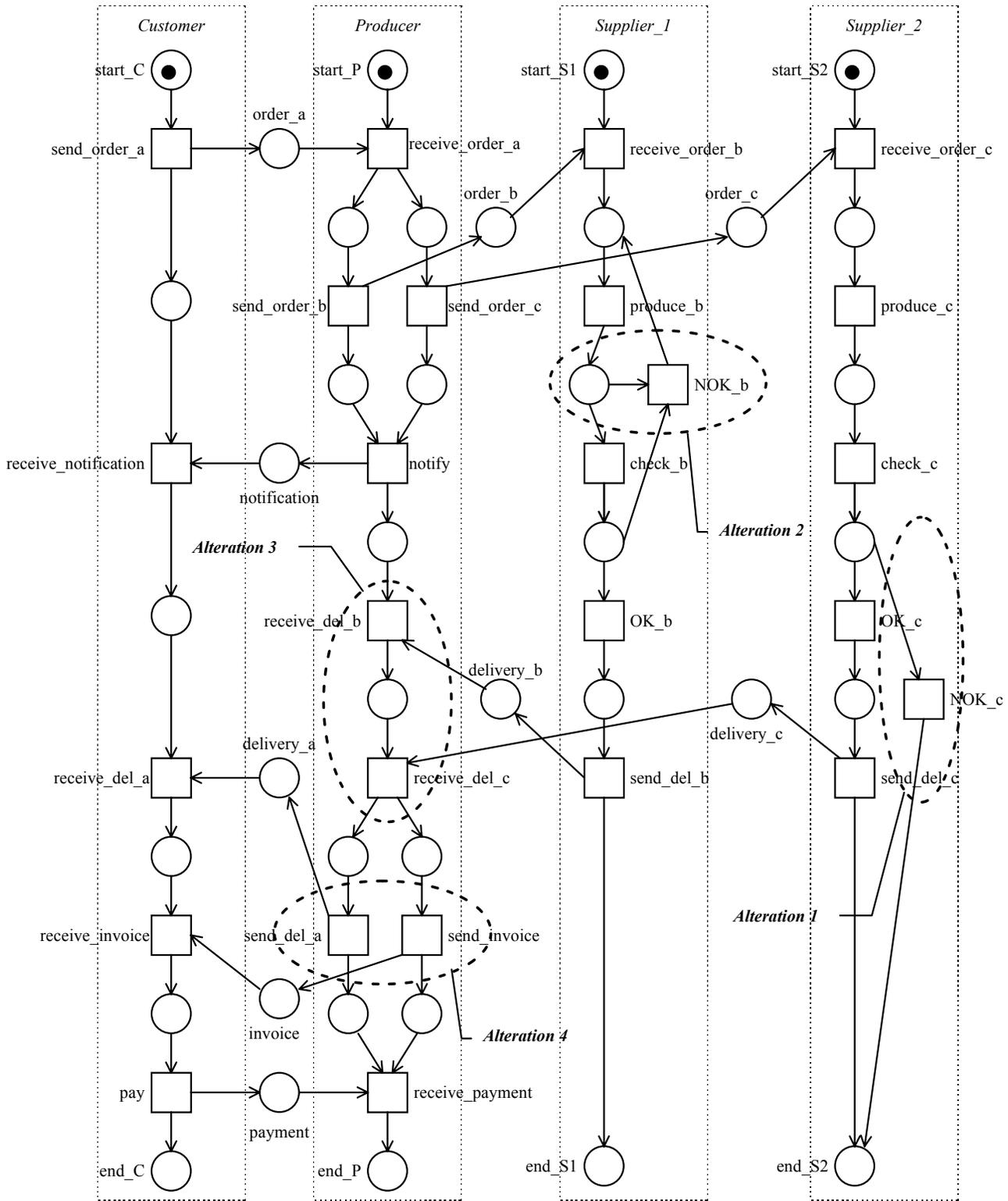
These changes are used to introduce two notions of correctness: *soundness* and *consistency*.

### 6.1. Soundness

A WF-net is sound if and only if it satisfies the following requirement. *For any case, the procedure will terminate eventually and the moment the procedure terminates there is a token in the output place and all the other places are empty.* Moreover, there should be no dead tasks, i.e., it should be possible to execute an arbitrary task by following the appropriate route though the WF-net. A formal definition is given in [1].

An interorganizational workflow is *locally sound* if and only if each of the local WF-nets in isolation is sound. Note that the WF-net which corresponds to *Supplier\_1* is not sound because *NOK\_b* is dead. The other three WF-nets in isolation are sound.

Given an interorganizational workflow, it is possible to construct an extended net by adding a new source (i.e., a global input place) connected to a transition which puts a token in each of the input places of the local WF-nets and a new sink (i.e., a global output place) connected to a transition which consumes a token from each of the output places of the local workflows. Note that the extended net is a WF-net [2,3]. An interorganizational workflow is *globally sound* if the extended net is sound. Clearly, alteration 2 invalidates global soundness. Although the local WF-net *Supplier\_2* is sound, alteration 1 also invalidates global soundness: If *NOK\_c* fires, then there is a deadlock in *Producer*.



**Figure 3: An alternative interorganizational workflow.**

It is also possible to have an interorganizational workflow that is globally sound but not locally and vice-versa. Therefore, we require an interorganizational workflow to be both globally sound and locally sound. The interorganizational workflow shown in Figure 2 has this property. Moreover, alterations 3 and 4 in Figure 3 do not invalidate soundness.

In [3] a decision procedure is given to decide soundness. This procedure uses state-of-the-art Petri-net-based analysis techniques. In fact it uses the fact that soundness corresponds to two well-known properties: liveness and boundedness [5,13]. For an organizational workflow composed of  $n$  local workflows we need to prove liveness and boundedness for  $n+1$  WF-nets ( $n$  local WF-nets and the extended net). We have developed a tool, named *Woflan* [15], to verify soundness. Woflan can interface with several workflow products (e.g., COSA, Staffware, METEOR, Protos) and can be downloaded via the World-Wide-Web (<http://www.win.tue.nl/~woflan/>).

## 6.2. Consistency

MSCs can be used to specify the interaction between loosely coupled workflow processes. An MSC can be used as a starting point for the design of complex interorganizational workflows. The interorganizational workflow should be designed in such a way that it is consistent with the MSC; i.e., it can be seen as a partial specification of an interorganizational workflow. Therefore, it is interesting to be able to decide whether the implementation (interorganizational workflow) meets the specification (MSC).

An interorganizational workflow is *consistent* with respect to an MSC if all possible firing sequences satisfy the partial order specified by the MSC. This means that the behavior of the interorganizational workflow may be more restrictive than the MSC, but it is not consistent if the behavior is extended. The interorganizational workflow shown in Figure 2 is consistent with the MSC shown in Figure 1. Alteration 3 in Figure 3 does not invalidate consistency. Alteration 4 invalidates consistency because it allows for a firing sequence (the invoice is send before the delivery) which is excluded in Figure 1. A formal definition of consistency is given in [3].

To check whether an interorganizational workflow is consistent with respect to an MSC one could inspect whether all possible firing sequences satisfy the partial order specified by the MSC. From a computational point of view, it is very expensive to check consistency this way.

We have developed an alternative technique for a restricted class of interorganizational workflows [3]. Instead of checking all possible firing sequences, we use a technique based on a generalization of the notion of *implicit places* [4]. An implicit or redundant place always contains sufficient tokens to allow for the firing of the transitions connected to it. The technique works as follows. Transitions in a restricted interorganizational workflow are associated with events. An MSC specifies a partial order on these events. Therefore, the MSC indirectly specifies a partial order on transitions. This partial order can be expressed in terms of places connecting transitions. These additional places are implicit, if and only if, the interorganizational workflow is consistent with respect to the MSC. This result is interesting because it allows for the efficient verification of consistency. In fact, for most of the problems encountered in practice consistency can be verified in polynomial time.

## 7. CONCLUSION

Many enterprises are involved in interorganizational workflows. These must be correct because problems resulting from errors in the design are difficult to trace and expensive to repair. Therefore, we propose a rigorous approach to the modeling and analyzing of workflows crossing organizational boundaries. This approach uses MSCs to *specify* the interaction between the business partners and Petri nets to *design* interorganizational workflows. We also discussed a basic property that any interorganizational workflow should satisfy: soundness. This property coincides with well-known Petri net properties. To establish the correctness of an interorganizational workflow composed of  $n$  local workflows, we need to prove liveness and boundedness for  $n+1$  WF-nets using standard techniques. Woflan, our workflow verification tool [15], uses state-of-the-art analysis techniques to decide soundness and to generate to-the-point diagnostic information.

In most cases, the interorganizational workflow should obey a given communication structure in addition to the soundness property. Given an MSC that specifies the communication between business partners, it is possible to use an efficient technique to verify whether the interorganizational workflow is consistent with the MSC.

## REFERENCES

1. W.M.P. van der Aalst. The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers*, 8(1):21-66, 1998.

2. W.M.P. van der Aalst. Modeling and Analyzing Interorganizational Workflows. In L. Lavagno and W. Reisig, editors, *Proceedings of the International Conference on Application of Concurrency to System Design (CSD'98)*, pages 1-15, IEEE Computer Society Press, 1998.
3. W.M.P. van der Aalst. Interorganizational Workflows: An Approach based on Message Sequence Charts and Petri Nets. *Systems Analysis - Modelling - Simulation*, 34(3):335-367, 1999.
4. G. Berthelot. Checking Properties of Nets Using Transformations. In G. Rozenberg, editor, *Advances in Petri Nets 1985*, volume 222 of *Lecture Notes in Computer Science*, pages 19-40. Springer-Verlag, Berlin, 1986.
5. J. Desel and J. Esparza. *Free choice Petri nets*, volume 40 of *Cambridge tracts in theoretical computer science*. Cambridge University Press, Cambridge, 1995.
6. C.A. Ellis and G.J. Nutt. Modeling and Enactment of Workflow Systems. In M. Ajmone Marsan, editor, *Application and Theory of Petri Nets 1993*, volume 691 of *Lecture Notes in Computer Science*, pages 1-16. Springer-Verlag, Berlin, 1993.
7. ITU-TS. ITU-TS Recommendation Z.120: Message Sequence Chart 1996 (MSC96). Technical report, ITU-TS, Geneva, 1996.
8. S. Jablonski and C. Bussler. *Workflow Management: Modeling Concepts, Architecture, and Implementation*. International Thomson Computer Press, 1996.
9. R. Kalakota and A.B. Whinston. *Frontiers of Electronic Commerce*. Addison-Wesley, Reading, Massachusetts, 1996.
10. P. Lawrence, editor. *Workflow Handbook 1997, Workflow Management Coalition*. John Wiley and Sons, New York, 1997.
11. S. Mauw and M.A. Reniers. An algebraic semantics of Basic Message Sequence Charts. *The Computer Journal*, 37(4):269-277, 1994.
12. M. Merz, B. Liberman, K. Muller-Jones, and W. Lamersdorf. Interorganisational Workflow Management with Mobile Agents in COSM. In *Proceedings of PAAM96 Conference on the Practical Application of Agents and Multiagent Systems*, 1996.
13. T. Murata. Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77(4):541-580, April 1989.
14. E. Rudolph, J. Grabowski, and P. Graubmann. Tutorial on Message Sequence Charts. *Computer Networks and ISDN Systems*, 28(12):1629-1641, 1996.
15. E. Verbeek and W.M.P. van der Aalst. *Woflan Home Page*, <http://www.win.tue.nl/~woflan>, 1999.

#### **BRIEF VITA**

**Wil van der Aalst** is an associate professor of Computing Science of Eindhoven University of Technology. He directs a small research group named Specification and Modeling of Information Systems (SMIS) and his current research interests include information systems, simulation, Petri nets, process models, workflow management systems, verification techniques, enterprise resource planning systems, computer supported cooperative work, and interorganizational business processes. Currently, Wil van der Aalst is a visiting professor at the University of Colorado.