Appendix A
The Order Fulfillment Process Model

Marcello La Rosa, Stephan Clemens, and Arthur ter Hofstede

A.1 Introduction

This appendix describes the Order Fulfillment process followed by a fictitious company named Genko Oil. The process is freely inspired by the VICS (Voluntary Inter-industry Commerce Solutions) reference model[1] and provides a demonstration of YAWL's capabilities in modeling complex control-flow, data, and resourcing requirements.

A.2 Overall Process

The Genko Oil company features four departments: the *Order Management Department* (OD), the *Supply Department* (SD) including the *Warehouse*, the *Carrier Department* (CD), and the *Finance Department* (FD). The Order Fulfillment process model is divided into the following phases:

- Ordering
- Logistics, which includes
  - Carrier Appointment
  - Freight in Transit
  - Freight Delivered
- Payment

The Order Fulfillment process model is shown in Fig. A.1, where each of the above phases is captured by a composite task. The orders remitted by customers

[1] www.vics.org

M. La Rosa
Queensland University of Technology, Brisbane, Australia,
e-mail: m.larosa@qut.edu.au

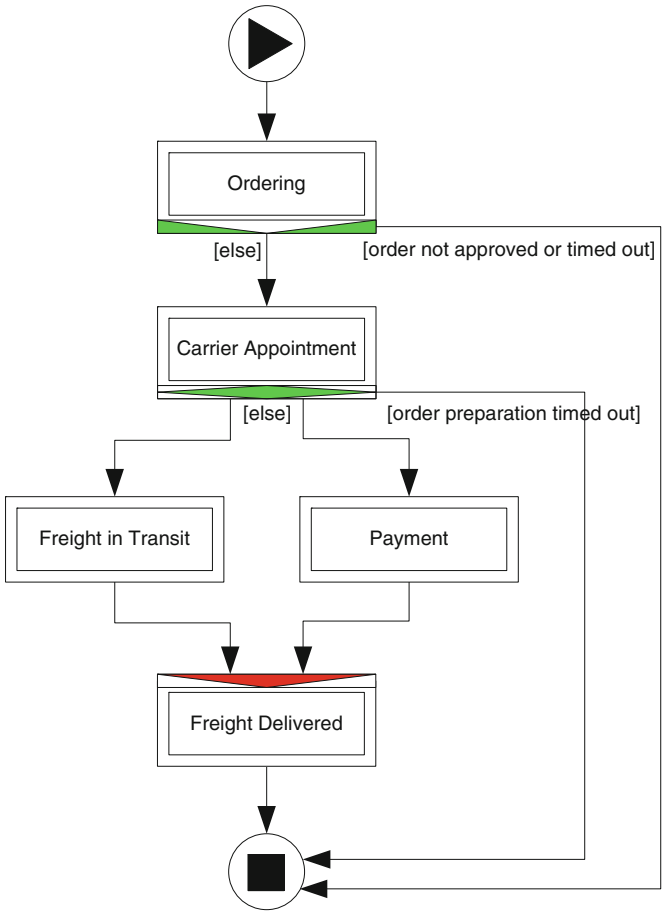

Fig. A.1 The Order Fulfillment process model in YAWL

are processed by the Orders Management Department. To keep the example manageable, a number of simplifying assumptions were made. For example, we assume that an order does not lead to more than one shipment. On the other hand, orders from different clients may be combined into a single shipment. A single package occupies only a fraction of a truck and can correspond to one of a fixed number of sizes.

The process starts with the *Ordering* task, where purchase orders can be created, modified, approved, or rejected. If an order is rejected or not confirmed in time, the process terminates, otherwise the logistical arrangements start in the task *Carrier Appointment*. This task deals with the preparation of the shipment quote, with delivery and pick-up arrangements and culminates in the actual pickup of the freight from the *Carrier Department*. If the quote is not prepared in time, the whole process terminates and the order is canceled. If the freight is picked up, tasks *Payment* and *Freight in Transit* are started in parallel. The former deals with the processing of the payment for the freight and for the shipment. The latter allows the Order department to issue inquiries after the status of the freight in transit, and handles notifications of the shipment's progress from the *Carrier Department*. This task terminates with the actual delivery of the freight to the customer. After completion of payment and

Table A.1 Participants of the Order Fulfillment process – name, position and role

User id	First name	Last name	Position	Role
ao	Arturo	de Ofstede	FD clerk	Finance Officer
bva	Billy	Van Arsdale	SD clerk	Junior Supply Officer
cc	Carmine	Cuneo	CD clerk	Shipment Planner
ccr	Connie	Corleone Rizzi	Head of warehouse	Warehouse Admin Officer
cm	Carmine	Marino	OD clerk	PO Manager, Client Liaison
cmc	Captain	McCluskey	FD clerk	Finance Officer
cr	Carlo	Rizzi	SD clerk	Junior Supply Officer
dcc	Don Carmine	Cuneo	FD clerk	Account Manager
dvc	Don Vito	Corleone	CEO	Order Fulfillment Manager
eb	Emilio	Barzini	FD clerk	Finance Officer
fc	Fredo	Corleone	Head of OD	PO Manager
jf	Johnny	Fontaine	CD clerk	Shipment Planner
jj	Jaggy	Jovino	SD clerk	Senior Supply Officer
jl	Joe	Lucadello	SD clerk, Warehouse clerk	Senior Supply Officer
jw	Jack	Woltz	Warehouse clerk	Warehouse Officer
ka	Kay	Adams	Head of CD	Carrier Admin Officer
mac	Mama	Corleone	CD clerk	Shipment Planner
mb	Momo	Barone	CD clerk	Courier
mc	Michael	Corleone	OD clerk	PO Manager
mlr	Marcello	La Rosa	CD clerk	Courier
pc	Peter	Clemenza	CD clerk	Courier
sc	Sonny	Corleone	OD clerk	PO Manager
sca	Stefano	Clemenza	CD clerk	Shipment Planner, Courier
st	Sal	Tessio	OD clerk	Client Liaison
th	Tom	Hagen	Head of SD, Assistant head of OD	Senior Supply Officer, Supply Admin Officer
vmc	Vincent 'Vinnie'	Mancini-Corleone	OD clerk	PO Manager
vs	Virgil 'The Turk'	Sollozzo	Head of FD	Senior Finance Officer

delivery, task *Freight Delivered* handles loss or damage claims and requests for return of merchandise. If no claim or request is lodged within a certain time frame, the process terminates.

Table A.1 lists all participants of the Order Fulfillment process with their user identifier, name, positions and roles. Each participant has default password "apple", whereas the YAWL administrator has user identifier "admin" and password "YAWL". Figure A.2 depicts the organizational chart of Genko Oil, where each participant belongs to one or more positions within a department.

All participants except *Don Vito Corleone* have the privilege to choose which work item to start, to start work items concurrently, and to reorder work items. *Don Carmine Cuneo* can also chain work item execution, and *Tom Hagen* and *Virgil "The Turk" Sollozzo* can also view all work items of their organizational group. *Don Vito Corleone* can only manage cases in his role of Manager of the Order Fulfillment process.

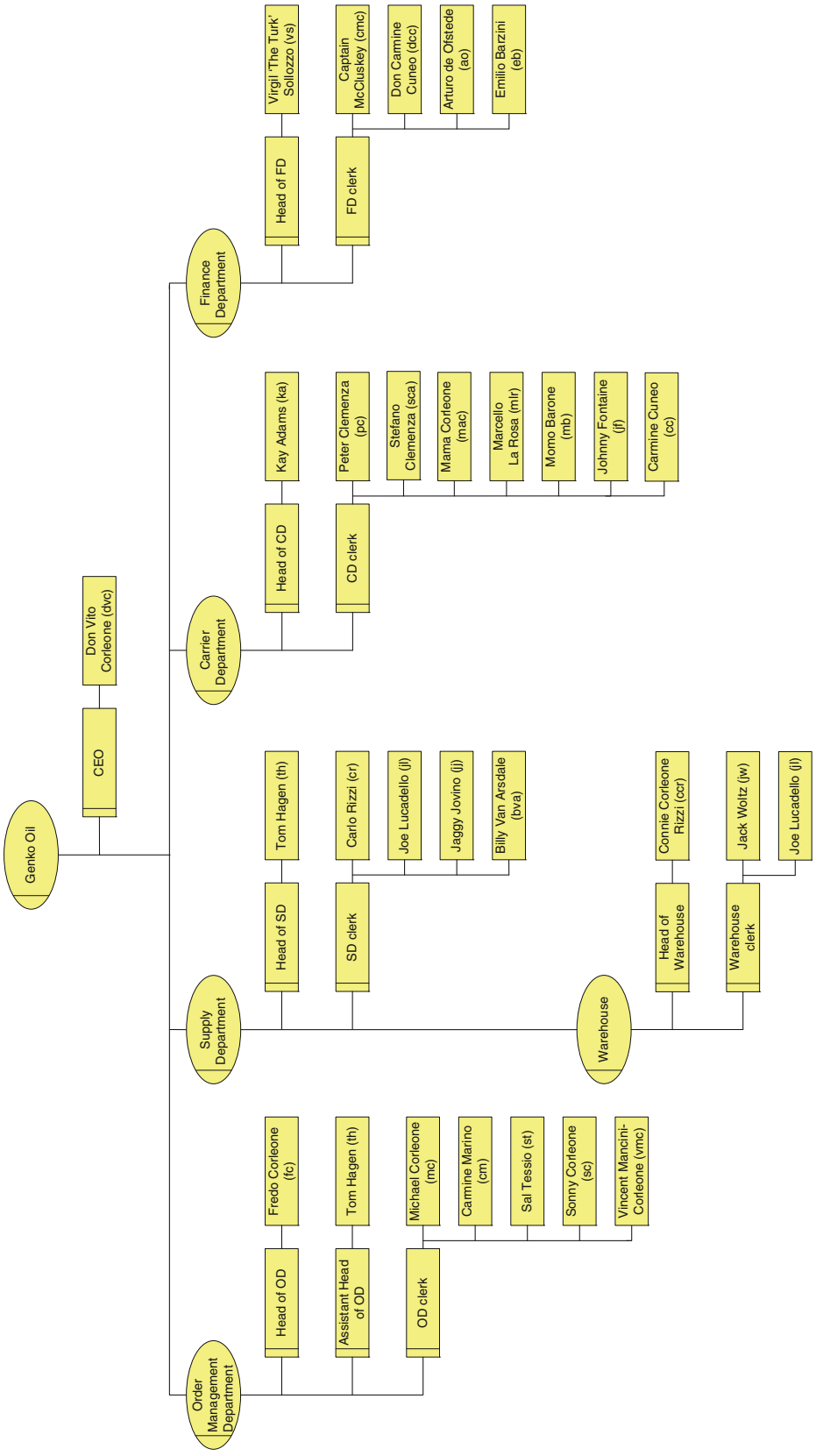

Fig. A.2 Organizational chart of the Genko Oil company

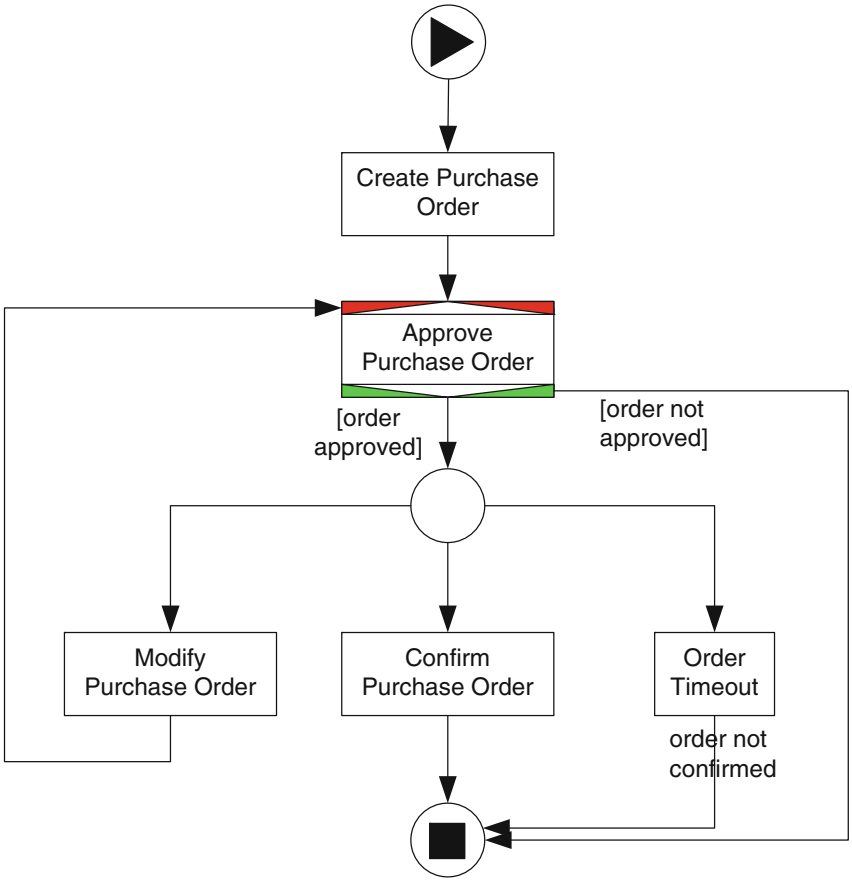

Fig. A.3 The *Ordering* subprocess

A.3 Ordering

The *Ordering* subprocess starts with the creation of a Purchase Order (PO) by the Order Management Department (see Fig. A.3). A PO needs to be approved by the *Supply Department* and may then be subject to a number of modifications, though it requires confirmation within a certain time frame.

The creation of a PO is handled by an *OD clerk*, who may choose to reallocate the task to another *PO Manager* with or without the work performed on it thus far. A *PO Manager* may also choose to relinquish working on the task and have the system offer it again to the available *PO Managers*. Moreover, a *PO Manager* may suspend working on the creation of a PO and choose to resume working on it at some later stage. Finally, a *PO Manager* may volunteer to be the main entry point for processing POs during a certain period of time. When a *PO Manager* is offered a task to create a PO, and they volunteer for it, the system will initiate the task automatically. Upon completing the PO, the *PO Manager* needs to decide which *PO Manager* will work on modification requests as they may eventuate at a later stage. The default *PO Manager* for PO modifications and confirmations is *Carmine Marino* (user id “cm”).

The completed PO is passed on to a *Supply Officer* who needs to approve it. If the *Supply Officer* who allocates this task to themselves is the *Head of SD*, they may choose to delegate this task to an *SD clerk* who reports to them. *Supply Officers* choose which approval tasks they will work on, and once they have chosen such a task, they may decide when to actually start working on it. This interaction pattern with the system is the default one for the various tasks that need to be performed in the Genko Oil company.

A PO contains information about the client's company (e.g., name, address and business number), information about the order (e.g., order number and date, currency, order terms, line items), the freight cost, and the delivery location. Moreover, it is possible to specify whether the order needs to be invoiced and whether it is part of a prepayment agreement between the client and Genko Oil.

Once a PO has been approved, repeated modifications may be requested. These are tracked by a revision number attached to the PO, which is increased at each change. Each of these changes again need to be approved. If the original PO or any modification is rejected, the order process ends. Moreover, the PO needs to be confirmed within 3 days, otherwise it is discarded and the process terminates.

A.4 Carrier Appointment

After confirmation of a PO, a route guide needs to be prepared and the trailer usage needs to be estimated. The route guide is prepared by determining the trackpoints that are going to be visited during the shipment. The trailer usage is determined by estimating the number of packages for the shipment, where each package has an identifier and a fixed volume of 25, 50, 100, or 200 lbs. These operations are performed in parallel by the two tasks *Prepare Route Guide* and *Estimate Trailer Usage* (see Fig. A.4). The former task is allocated to the Shipment Planner with the shortest work queue, while the latter task is allocated to the Shipment Planner who was allocated an instance of this task the longest time ago. If either task takes too long, a time out is triggered, which leads to the cancelation of the PO and the termination of the overall process. This timer is set to 5 days for a PO with one line item and is increased by 1 day for each additional line item. This calculation is performed by the automated task *Calculate Carrier Timeout*, which is assigned to a codelet to perform the required additions.

If both tasks *Prepare Route Guide* and *Estimate Trailer Usage* are completed in time, a *Supply Officer* can perform the task *Prepare Transportation Quote* by establishing the shipment cost based on the number of packages and on the total volume of the freight, and by assigning a shipment number to the order number. Once a *Supply Officer* has chosen to perform this task, they have the privilege to reallocate it to someone else without loss of the work performed thus far, and to suspend and resume working on the task. In addition, the system automatically starts the task for a *Supply Officer* once they have committed themselves to performing it.

After the task *Prepare Transportation Quote*, based on the total volume of the freight and on the number of packages, a distinction is made among shipments that

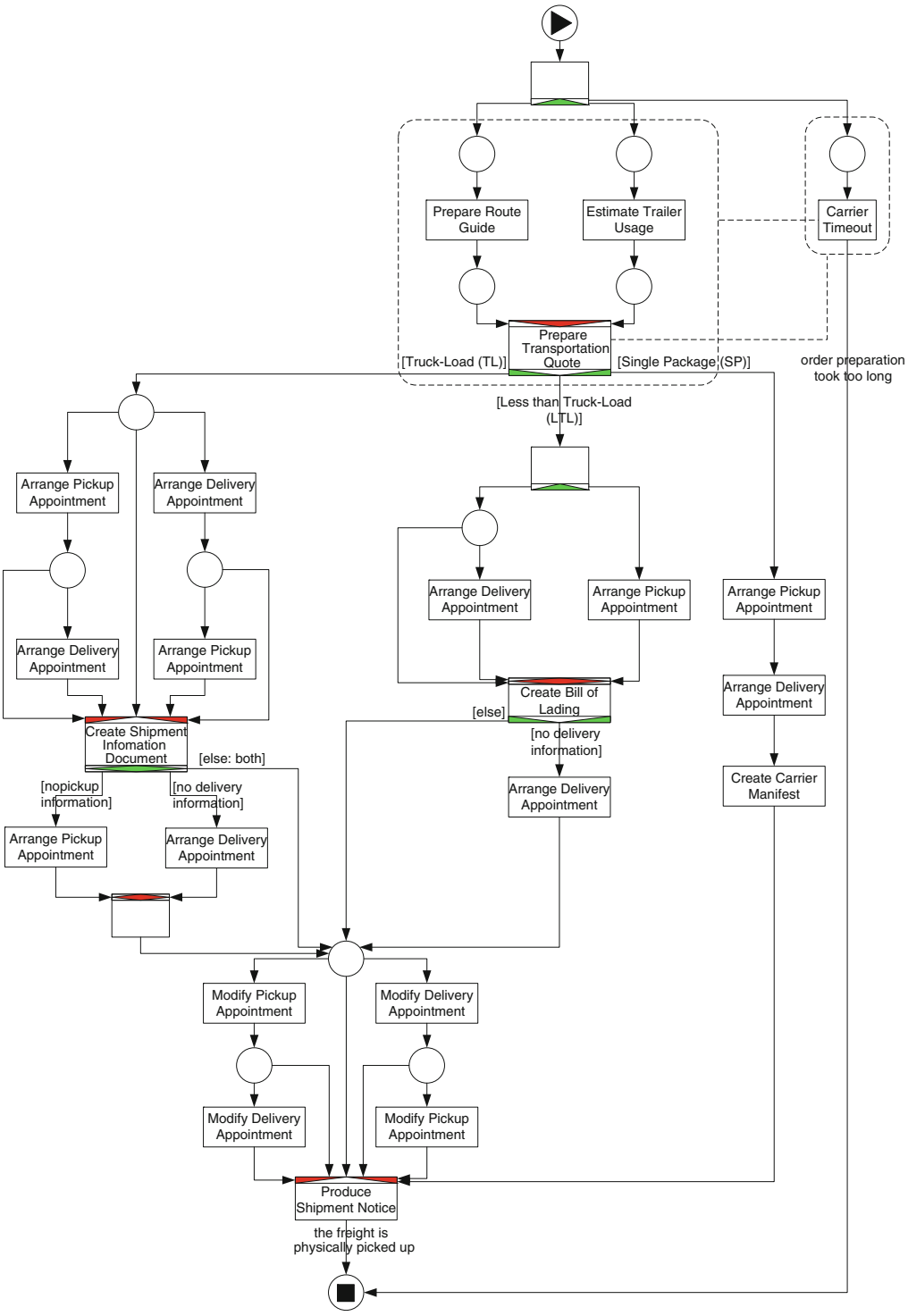

Fig. A.4 The *Carrier Appointment* subprocess

require a full truck load (total volume greater than or equal to 10.000 lbs), those that require less than a truck load (total volume less than 10.000lbs and more than one package), and those that simply concern a single package (total volume less than 10.000 lbs). For shipments that require a full truck load, *Client Liaisons* from the *OD* try to arrange a Pickup appointment and a Delivery appointment by specifying the location for pickup/delivery and any specific instructions. The *Client Liaisons* associated with these two tasks should be different. It is possible that only one of these or even none of these appointments is made before a *Senior Supply Officer* holding a *Master's in Supply Chain and Logistics Management* decides to create a Shipment Information document.

The Shipment Information document is used by the *Senior Supply Officer* to specify an authorization code and a consignee number for the shipment number. After the creation of this document, any missing appointments are made, though at this time a *Warehouse Officer* takes charge of arranging a Pickup appointment and a *Supply Officer* takes care of arranging a Delivery appointment, and there are subsequent opportunities to modify them until a *Warehouse Admin Officer* produces a Shipment Notice after which the freight can actually be picked up from the *Warehouse*. Modifications of Pickup appointments are handled by a *Warehouse Officer*, while modifications of Delivery appointments are taken care of by a *Supply Officer*.

When the shipment consists of more than one package but a dedicated truck is not required, then a *Warehouse Officer* arranges a Pickup appointment and a Client Liaison tries to arrange a Delivery appointment. Afterwards, a *Senior Supply Officer*, who holds a *Bachelor's in Supply Chain and Logistics Management*, creates a Bill of Lading, which, similar to the Shipment Information document, requires the specification of an authorization code and a consignee number. If no Delivery appointment was made prior, a *Supply Officer* takes care of this and the remainder of the process is the same as for a shipment that requires a dedicated truck.

For shipments consisting of a single package, the process is straightforward. All that needs to be done is by a *Supply Officer* – one who has the most experience in performing this particular task (identified using the "Round Robin by Experience" allocation strategy) – to create a Motor Carrier Pickup manifest. This is done by specifying only an authorization code. Afterwards, a Shipment Notice is produced by a *Warehouse Admin Officer* and the freight is ready for pickup.

A Shipment Notice provides a summary of the shipment, which includes the shipment number, the order number, the number of packages, the pickup and delivery appointments, and a variable indicating whether the shipment is a full truck load. The *Warehouse Admin Officer* has to indicate when the freight loading on the carrier's truck started and completed and provide the details of the driver (number and name), the delivery instructions, and a deadline for the claims, which will be used later on in the subprocess *Freight Delivered*. Delivery instructions contain textual instructions, the delivery date, and the delivery location (the latter being retrieved from the Route Guide).

A.5 Payment

After a freight has been picked up, the *Payment* subprocess can start (see Fig. A.5). This process has two components, one which is concerned with payment for the shipment and one which is concerned with payment for the freight.

The first task that needs to be performed in dealing with the payment for a shipment is the production of a Shipment Invoice containing the shipment costs related to the order number, the shipment number, and the company to be invoiced. This task is handled by a *Supply Admin Officer*. During the creation of the PO in the *Ordering* subprocess, a *PO Manager* specified if shipments were paid in advance. In this case, all that is required is for a *Finance Officer* to issue a Shipment Remittance Advice, where he/she specifies the amount being debited to the client. Otherwise, a *Finance Officer* issues a Shipment Payment Order. This document includes a shipment payment order number that refers to both the order and the shipment numbers, the shipment costs, and the details of the beneficiary, which includes the company, the name and code of the beneficiary bank, and the name and number of the bank account.

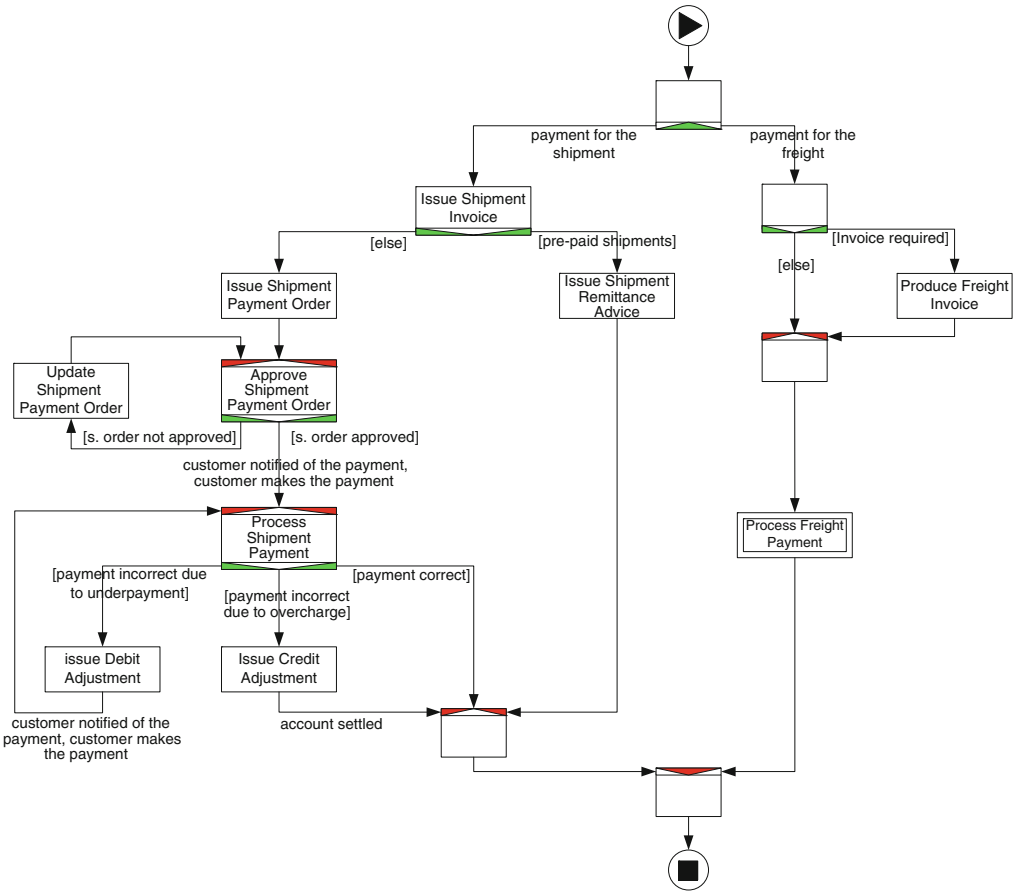

Fig. A.5 The *Payment* subprocess

A *Senior Finance Officer* who is the superior of this *Finance Officer* has to approve this document. A number of updates may be made to this document by the *Finance Officer* who issued it, but each of these need to be approved by the *Senior Finance Officer*. After the document is finalized, an *Account Manager* can process the payment for the shipment once the client has made the payment, by specifying the balance. It is possible that the client has underpaid. This case requires a debit adjustment (the amount still due is specified), the Account Manager needs to notify the client and the payment needs to be reprocessed. A client can also overpay and then the *Account Manager* needs to perform a credit adjustment (the reimbursement amount and the beneficiary details are specified). In the latter case and in case of correct payment, the shipment payment process is completed. An *Account Manager* can choose to have adjustment tasks to be started immediately for them upon completion of the processing of a Shipment Payment in order to expedite matters.

As regards the payment for a freight, a *PO Manager* specifies during creation of a PO whether an invoice is required or not. Should an invoice be required, a Freight Invoice is created by a *Supply Admin Officer*, containing the freight costs related to the order number and the company to be invoiced. If a Product Invoice is not required, then processing of the freight payment can start right away, if not, it can begin after the creation of the Freight Invoice. The processing of a freight payment is an involved subprocess, which we will not consider further.

A.6 Freight in Transit

The *Freight in Transit* subprocess is performed in parallel with the *Payment* subprocess and is concerned with tracking progress of the delivery of an order and with handling client inquiries (see Fig. A.6).

A delivery truck may visit multiple destinations and some of these may be designated trackpoints that may assist in finding out where a certain shipment is. When a trackpoint is visited, a Trackpoint Notice is issued by the *Courier*, registering the truck's arrival and departure time, plus additional notes. The *Courier* may choose to skip this task as it may lead to undesirable delay in some cases.

Once all trackpoints have been visited, *Carrier Admin Officers* need to log a Trackpoint Order Entry for each trackpoint, which contains a report for the specific trackpoint so that this information can be audited in the future. Trackpoint Order Entries can be logged in parallel so long as there are Carrier Admin Officers available to work on the entries. While entries are being logged, customer enquiries need to be addressed by a *Client Liaison*. Once completed, new enquiries can be ignored and a *PO Manager* creates an Acceptance Certificate to register that the freight has been physically delivered. An Acceptance Certificate refers to an order number and a shipment number and contains the acceptance date and delivery notes.

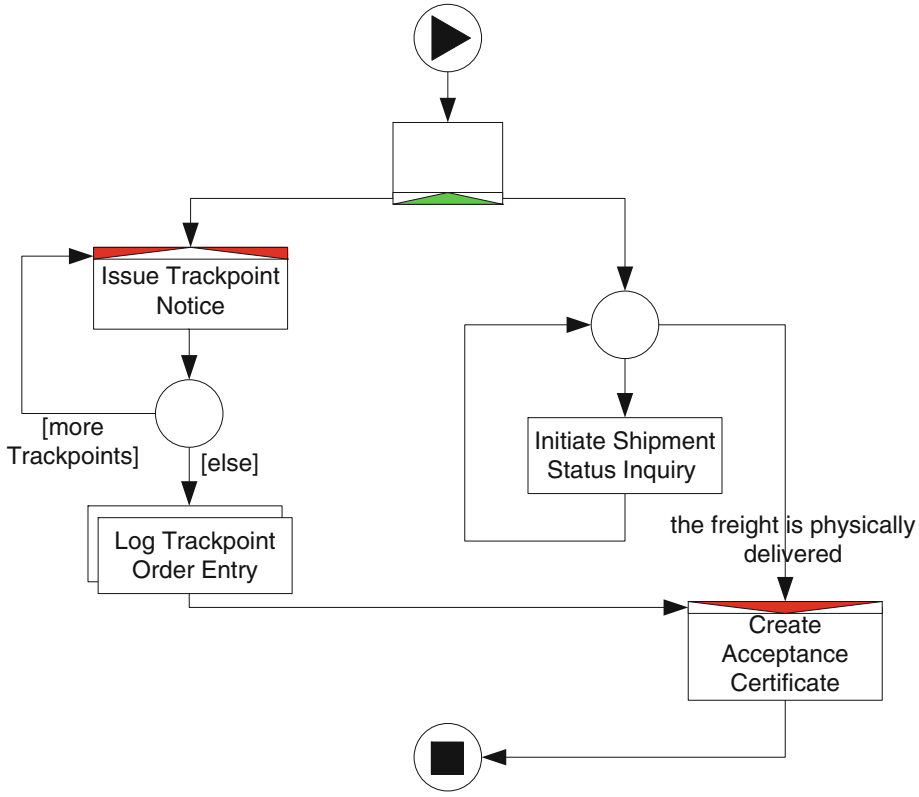

Fig. A.6 The *Freight in Transit* subprocess

A.7 Freight Delivered

After the delivery of the freight, clients may request a Return of Merchandise or may want to lodge a Loss or Damage Claim (see Fig. A.7). A Return of Merchandise document contains the reason for return and the line items to be returned, while a Loss or Damage Claim indicates the reason for claim. Both these documents are lodged into the system by a *Client Liaison* upon the client's request. If no client communications are received within a certain period after the delivery – the length of which is specified in the Shipment Notice – the Order Fulfillment process is considered to be successfully completed.

The request for a Return of Merchandise needs to be approved by a *Senior Supply Officer*. If approved, an involved Return Management process starts, which will not be considered further. Similarly, a Loss or Damage Claim also needs to be approved by a *Senior Supply Officer*, and if approved, will involve the process of actually managing this Loss or Damage Claim. Again, this is something we will not consider any further. In either case, if approval is not granted, the process is considered to be completed.

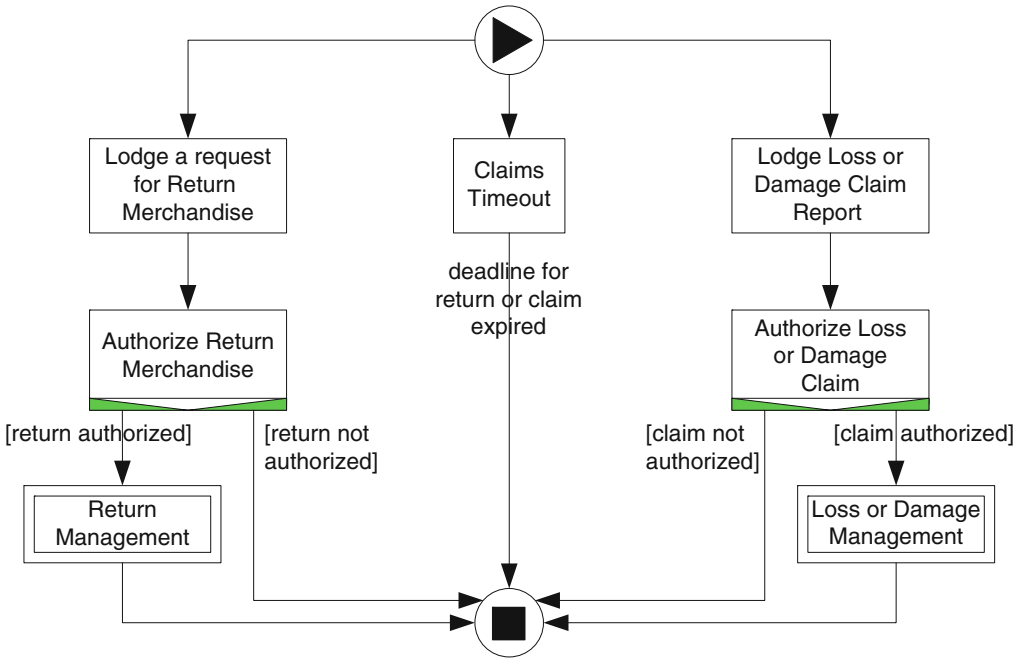

Fig. A.7 The *Freight Delivered* subprocess

A.8 Showcased YAWL features

The Order Fulfillment process demonstrates a wide range of language features. This section illustrates some advanced ones.

Cancelation Region: The subprocess *Carrier Appointment* contains two cancelation sets. The first one is triggered by the timer task *CarrierTimeout*, which, once completed, leads to the termination of the Order Fulfillment process by consuming tokens in tasks *Prepare Route Guide*, *Estimate Trailer Usage*, *Prepare Transportation Quote*, and in the conditions in-between. Similarly, the second cancelation set is triggered after completion of task *Prepare Transportation Quote*, leading to the consumption of tokens in the task *CarrierTimeout* and in its preceding condition.

Multiple Instance Task: In subprocess *Freight in Transit*, multiple instance task *Log Trackpoint Order Entry* reads all Trackpoint Notices created in task *Issue Trackpoint Notice* and assigns each notice to a *Carrier Admin Officer* who can attach a report to it. Multiple *Carrier Admin Officers* can work in parallel on this task.
OR-join: Subprocess Carrier Appointment features two OR-join constructs. The first one is in the truck load branch after tasks *Arrange Pickup Appointment* and *Arrange Delivery Appointment*. The second OR-join is attached to task *Create Bill of Lading* in the less than truck load branch.

Timer Task: Subprocess Carrier Appointment features a timer task that fires upon work item starting (task *Carrier Timeout*), while subprocess *Freight Delivered* features a timer task that fires on work item enablement (task *Claims Timeout*). The deadline associated with both these timer tasks is dynamically determined via a net variable.

Codelet: Automated task *Calculate Carrier Timeout* (subprocess *Carrier Appointment*) determines the timeout for the preparation of the Transportation Quote based on the number of Order Lines in the PO. This is done by using the *ExampleCodelet* in YAWL, which allows one to make simple additions.

Extended attributes: In task *Prepare Route Guide* of subprocess *Carrier Appointment*, task variables `OrderNumber` and `DeliveryLocation` are both Input and Output because their values need to be shown to the user and then used to compose the content of net variable `Prepare_Route_Guide`. However, to avoid that these values are modified by the user, the two variables have been assigned extended attribute `readOnly` with value `true`.

Dynamic Task Allocation: In subprocess *Ordering*, task *Modify Purchase Order* is allocated to the participant whose user id is stored in net variable `PO_Manager`. This variable is set by task *Create Purchase Order* before executing *Modify Purchase Order*.

Retain Familiar: When a single package is shipped, the participant who arranges the pickup appointment (task *Arrange Pickup Appointment* of subprocess Carrier_Appointment) has also to arrange the Delivery Appointment (task *Arrange Delivery Appointment*).

Four Eyes Principle: In full truck load shipments, tasks *Arrange Delivery Appointment* and *Arrange Pickup Appointment* of subprocess *Carrier Appointment* are never offered to the same participant during the same process instance.

Distribution Set Filter: The initial distribution set assigned to task *Create Purchase Order* is filtered such that this task can be piled to a single participant (runtime filtering). Similarly, a filter on capabilities is applied to the distribution set assigned to tasks *Create Shipment Information Document* (which is offered only to a *Senior Supply Officer* holding a *Master's in Supply Chain and Logistics Management*) and *Create Bill of Lading* (which is only offered to a *Senior Supply Officer* holding a *Bachelor's in Supply Chain and Logistics Management*).

Allocation Strategy: In subprocess *Carrier Appointment*, task *Estimate Trailer Usage* is allocated to the Shipment Planner who was allocated this task the longest time ago (allocation by time). Similarly, in case of single package shipments, task *Arrange Pickup Appointment* is allocated to the *Supply Officer* who executed this task the most (allocation by experience).

Privileges: Any participant with role of *Account Manager* has the privilege to chain the execution of work items assigned to them. In this way an *Account Manager* can have tasks *Issue Debit Adjustment* and *Issue Credit Adjustment* of subprocess *Payment* immediately started for them upon completion of task *Process Shipment Payment*.

A.9 Setup

The Order Fulfillment process has been tested to be executed by any YAWL 2.0 engine release (YAWL4Study, YAWL4Enterprise, and YAWLive). This section provides instructions on how to successfully launch a case of the Order Fulfillment process.

First, import the organizational data associated with the Order Fulfillment example into the yawl database:

1. Start the YAWL Engine
2. Open the Control Centre and log in as administrator (id "admin", password "YAWL")
3. Click on *Org Data*
4. If the YAWL database has already been populated with resources, backup that data by clicking on the icon "Export Org Data to file" on the top-right corner
5. Click on the icon "Import Org Data from file" on the top-right corner and select the file *orderfulfillment.ybkp* from the directory *[YAWL Installation folder]/misc/examples/orderfulfillment/*

A message box at the bottom should appear and indicate that the operation has succeeded by listing the number of resources that have been imported in terms of participants, groups, capabilities, and positions.

Now, a case of the Order Fulfillment process may be launched:

1. Click on *Cases* in the Control Centre
2. Upload the file *orderfulfillment.yawl* from the directory *[YAWL Installation folder]/misc/examples/orderfulfillment/*
3. Select the loaded specification *OrderFulfillment*
4. Click on *Launch Case*

Log out as administrator and log in as an employee of Genko Oil to view and edit the first work item. For example, log in as Fredo Corleone (user id "fg," password "apple") and start the creation of a PO.

Appendix B
Mathematical Notation

Nick Russell

This appendix outlines mathematical notations used in this book, which are not in general use and hence merit some further explanation.

Given two sets X and Y, a function $f : X \to Y$ can be defined, indicating that for $x \in X$ there exists *exactly one* element $y \in Y$ such that $f(x) = y$. If f applies to *all* members of X, then f is considered to be a *total* function (generally denoted as $f : X \to Y$). If the function does not apply to all members of X, then f is considered to be a *partial* function and is denoted $f : X \nrightarrow Y$.

A function $f : X \to Y$ can be considered to be comprised of a set of ordered pairs $(x, y) \in f$, where $x \in X, y \in Y$, and $f(x) = y$. The domain restriction operator (dom) when applied to f yields the set of elements $x \in X$ that participate in the function, that is, $dom(f) = \{x \mid (x, y) \in f\}$.

In the context of a function $f : X \to Y$, range restriction of f over a set $R \subseteq X$ is defined by $f \rhd R = \{(x, y) \in f \mid y \in R\}$.

The difference between two sets A and B can be defined as $A \backslash B = \{x \in A \mid x \notin B\}$.

$\mathbb{P}(X)$ denotes the power set of X, where $Y \in \mathbb{P}(X) \Leftrightarrow Y \subseteq X$.

$\mathbb{P}^+(X)$ denotes the power set of X without the empty set, that is, $\mathbb{P}^+(X) = \mathbb{P}(X) \backslash \{\varnothing\}$.

Let $V = \{v_1, \ldots v_n\}$ be a (nonempty) set and $<$ a strict total order over V, then $[V]^<$ denotes the sequence $[v_1, \ldots v_n]$ such that $\forall_{1 \le i \le j < n}[v_i < v_j]$ and every element of V occurs precisely once in the sequence. $[V]$ denotes the sequence in arbitrary order. Sequence comprehension is defined as $[E(x) \mid x \leftarrow [V]^<]$, yielding a sequence $[E(v_1) \ldots E(v_n)]$.

N. Russell
Eindhoven University of Technology, Eindhoven, the Netherlands,
e-mail: n.c.russell@tue.nl

A bag is a finite multi-set of elements from a given alphabet A and can be considered to take the form of a function from A to the set of natural numbers $\mathbb{N}$. A bag only contains a finite number of elements from A, each of which is mapped to a non-zero value. For a given bag X comprised of elements from alphabet A where $a \in A$, $X(a)$ denotes the number of occurrences of a in A, a notion also known as the cardinality of a in X.

The explicit enumeration of a bag uses a similar notation to that for sets except that square brackets are used instead of curly brackets and superscripts identify the cardinality of individual elements. For example $[a^2, b^3, c]$ denotes the bag with two elements a, three elements b and one c. The bag $[a^2 \mid P(a)]$ contains two elements a such that for every element a function $P(a)$ holds where P is some predicate on the symbols of the alphabet A under consideration. To denote individual elements of a bag, the "$\in$" symbol is used as for sets. For any bag X over alphabet A and element $a \in A$, $a \in X$ if and only if $X(a) > 0$.

The sum of two bags X and Y, denoted $X + Y$, is defined as $[a^n \mid a \in A \wedge n = X(a) + Y(a)]$. The difference of X and Y, denoted $X - Y$, is defined as $[a^n \mid a \in A \wedge n = max((X(a) - Y(a)), 0)]$.

Appendix C
The Original Workflow Patterns

Nick Russell

Since the first group of 20 control-flow patterns were identified and documented in 1999–2000, the Workflow Patterns have been subject to ongoing scrutiny and enhancement. In 2006, a comprehensive review was conducted of the original control-flow patterns, which resulted in the collection being augmented with a further 23 new patterns. Some of these were specializations of existing patterns, while others recognized earlier gaps in the scope of the pattern set.

Table C.1 identifies the original control-flow patterns that were specialized into two or more patterns that more precisely reflect specific problem/solution scenarios that were previously amalgamated in the original pattern definitions.

See www.workflowpatterns.com for an up-to-date and detailed description of all workflow patterns.

Table C.1 Refinements of original control-flow patterns

Original patterns	Revised patterns
Synchronization	– Synchronization – Generalized AND-Join
Synchronizing Merge	– Structured Synchronizing Merge – Local Synchronizing Merge – General Synchronizing Merge
Discriminator	– Structured Partial Join – Local Partial Join – General Partial Join
Multiple Instances with a priori Design Time Knowledge	– Multiple Instances with a priori Design Time Knowledge – Static Partial Join for Multiple Instances – Canceling Partial Join for Multiple Instances

(continued)

N. Russell
Eindhoven University of Technology, Eindhoven, the Netherlands,
e-mail: n.c.russell@tue.nl

Table C.1 (continued)

Original patterns	Revised patterns
Multiple Instances without a priori Design Time Knowledge	– Multiple Instances without a priori Design Time Knowledge – Dynamic Partial Join for Multiple Instances
Interleaved Parallel Routing	– Interleaved Parallel Routing – Interleaved Routing – Critical Section
Cancel Activity	– Cancel Task – Cancel Region – Cancel Multiple Instance Task – Complete Multiple Instance Task

Table C.2 Retained and new control-flow patterns

Original patterns	Revised patterns
Retained patterns	
Sequence	Sequence
Parallel Split	Parallel Split
Exclusive Choice	Exclusive Choice
Simple Merge	Simple Merge
Multiple Choice	Multiple Choice
Multiple Merge	Multiple Merge
Arbitrary Cycles	Arbitrary Cycles
Implicit Termination	Implicit Termination
Multiple Instances without Synchronization	Multiple Instances without Synchronization
Multiple Instances with a priori Runtime Knowledge	Multiple Instances with a priori Runtime Knowledge
Deferred Choice	Deferred Choice
Milestone	Milestone
Cancel Case	Cancel Case
New Patterns	
	Thread Merge
	Thread Split
	Recursion
	Explicit Termination
	Structured Loop
	Transient Trigger
	Persistent Trigger

Table C.2 identifies those patterns that were essentially retained in the same format in the revised pattern set and also shows the new patterns that were introduced as a consequence of the review.

References

1. W.M.P. van der Aalst, Three good reasons for using a Petri-net-based workflow management system, ed. by S. Navathe, T. Wakayama. In *Proceedings of the International Working Conference on Information and Process Integration in Enterprises (IPIC'96)*, Cambridge, MA, USA, November 1996, pp. 179–201
2. W.M.P. van der Aalst. The application of Petri nets to workflow management. J. Circ. Syst. Comput. **8**(1), 21–66 (1998)
3. W.M.P. van der Aalst, Formalization and verification of Event-driven Process Chains. Inform. Software Tech. **41**(10), 639–650 (1999)
4. W.M.P. van der Aalst, Patterns and XPDL: A critical evaluation of the XML process definition language, BPM Center Report BPM-03-09, BPMcenter.org (2003)
5. W.M.P. van der Aalst, M. Adams, A.H.M. ter Hofstede, M. Pesic, M.H. Schonenberg, ed. by L. Chen, C. Liu, Q. Liu, Ke Deng. In Database Systems for Advanced Applications, DASFAA 2009 International Workshops: BenchmarX, MCIS, WDPP, PPDA, MBC, PhD, Brisbane, Australia, April 20–23, 2009, Lecture Notes in Computer Science, vol. 5667, Brisbane, Australia, April 2009 (Springer, Berlin, 2009), pp. 319–333
6. W.M.P. van der Aalst, L. Aldred, M. Dumas, A.H.M. ter Hofstede, Design and implementation of the YAWL system, ed. by A. Persson, J. Stirna. In *Proceedings of the 16th International Conference on Advanced Information Systems Engineering (CAiSE'04)*, Lecture Notes in Computer Science, vol. 3084, Riga, Latvia, June 2004 (Springer, Berlin, 2004), pp. 142–159
7. W.M.P. van der Aalst, A.P. Barros, A.H.M. ter Hofstede, B. Kiepuszewski, Advanced workflow patterns, ed. by O. Etzion, P. Scheuermann. In *Proceedings of the 5th IFCIS International Conference on Cooperative Information Systems (CoopIS'2000)*, Lecture Notes in Computer Science, vol. 1901, Eilat, Israel, September 2000 (Springer, Berlin, 2000), pp. 18–29
8. W.M.P. van der Aalst, T. Basten, Inheritance of workflows: An approach to tackling problems related to change. Theor. Comput. Sci. **270**(1–2), 125–203 (2002)
9. W.M.P. van der Aalst, J. Desel, E. Kindler, On the semantics of EPCs: A vicious circle, ed. by M. Rump, F.J. Nüttgens. In *Proceedings of the EPK 2002: Business Process Management using EPCs*, Trier, Germany, November 2002 (Gesellschaft für Informatik, Germany, 2002), pp. 71–80
10. W.M.P. van der Aalst, M. Dumas, F. Gottschalk, A.H.M. ter Hofstede, M. La Rosa, J. Mendling, Preserving correctness during business process model configuration. Formal Aspect. Comput. (2009). doi: 10.1007/s00165-009-0112-0
11. W.M.P. van der Aalst, K.M. van Hee, *Workflow Management: Models, Methods and Systems*, new edn. (MIT, Cambridge, MA, 2004)
12. W.M.P. van der Aalst, K.M. van Hee, A.H.M. ter Hofstede, N. Sidorova, H.M.W. Verbeek, M. Voorhoeve, M.T. Wynn, Soundness of workflow nets: Classification, decidability, and analysis, BPM Center Report BPM-08-02, BPMCenter.org (2008)
13. W.M.P. van der Aalst, K.M. van Hee, A.H.M. ter Hofstede, N. Sidorova, H.M.W. Verbeek, M. Voorhoeve, M.T. Wynn, Soundness of workflow nets with reset arcs is undecidable! ed. by J. Kleijn, M. Koutny. In *Proceedings of the International Workshop on Concurrency Methods*

Issues and Applications (CHINA'08), Xi'an, China, June 2008 (Xidian University, China, 2008), pp. 57–72
14. W.M.P. van der Aalst, A.H.M. ter Hofstede, Workflow patterns: On the expressive power of Petri-net-based workflow languages, ed. by K. Jensen. In *Proceedings of the 4th International Workshop on Practical Use of Coloured Petri Nets and the CPN Tools*, Aarhus, Denmark, August 2002, pp. 1–20, Technical Report DAIMI PB-560 (2002)
15. W.M.P. van der Aalst, A.H.M. ter Hofstede, YAWL: Yet Another Workflow Language. Inform. Syst. **30**(4), 245–275 (2005)
16. W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, A.P. Barros, Workflow patterns. Distr. Parallel Database **14**(3), 5–51 (2003)
17. W.M.P. van der Aalst, S. Jablonski, Dealing with workflow change: Identification of issues and solutions. Int. J. Comput. Syst. Sci. Eng. **15**(5), 267–276 (2000)
18. W.M.P. van der Aalst, J. Nakatumba, A. Rozinat, N. Russell, Business process simulation: How to get it right? BPM Center Report BPM-08-07, BPMcenter.org (2008)
19. W.M.P. van der Aalst, M. Pesic, DecSerFlow: Towards a truly declarative service flow language, ed. by M. Bravetti, M. Nunez, G. Zavattaro. In *Proceedings of the International Conference on Web Services and Formal Methods (WS-FM'06)*, Lecture Notes in Computer Science, vol. 4184, Vienna, Austria, 2006 (Springer, Berlin, 2006), p. 1–23
20. W.M.P. van der Aalst, M. Pesic, in *Specifying and Monitoring Service Flows: Making Web Services Process-Aware*, ed. by Baresi, Di Nitto. Test and Analysis of Web Services (Springer, Berlin, 2007), pp. 11–56
21. W.M.P. van der Aalst, A.J.M.M. Weijters, L. Maruster, Workflow mining: Discovering process models from event logs. IEEE Trans. Knowl. Data Eng. **16**(9), 1128–1142 (2004)
22. W.M.P. van der Aalst, M. Weske, D. Grünbauer, Case handling: A new paradigm for business process support. Data Knowl. Eng. **53**(2), 129–162 (2005)
23. M. Adams, *Facilitating Dynamic Flexibility and Exception Handling for Workflows*, PhD thesis, Queensland University of Technology, Brisbane, Australia, 2007
24. M. Adams, D. Edmond, A.H.M. ter Hofstede, The application of Activity Theory to dynamic workflow adaptation issues, ed. by J. Hanisch, D. Falconer, S. Horrocks, M. Hillier. In *Proceedings of the 2003 Pacific Asia Conference on Information Systems (PACIS 2003)*, Adelaide, Australia, July 2003 (University of South Australia, Australia, 2003), pp. 1836–1852
25. M. Adams, A.H.M. ter Hofstede, W.M.P. van der Aalst, D. Edmond, Dynamic, extensible and context-aware exception handling for workflows, ed. by R. Meersman, Z. Tari. In *Proceedings of the 15th International Conference on Cooperative Information Systems (CoopIS'07)*, Lecture Notes in Computer Science, vol. 4803, Vilamoura, Portugal, November 2007 (Springer, Berlin, 2007), pp. 95–112
26. M. Adams, A.H.M. ter Hofstede, D. Edmond, W.M.P. van der Aalst, Facilitating flexibility and dynamic exception handling in workflows through worklets, ed. by O. Bello, J. Eder, O. Pastor, J. Falcão e Cunha. In *Proceedings of the CAiSE'05 Forum*, FEUP edições, Porto, Portugal, June 2005, pp. 45–50
27. M. Adams, A.H.M. ter Hofstede, D. Edmond, W.M.P. van der Aalst, Worklets: A service-oriented implementation of dynamic flexibility in workflows, ed. by R. Meersman, Z. Tari et al. In *Proceedings of the 14th International Conference on Cooperative Information Systems (CoopIS'06)*, Lecture Notes in Computer Science, vol. 4275, Montpellier, France, November 2006 (Springer, Berlin, 2006), pp. 291–308
28. A. Agrawal, M. Amend, M. Das, M. Ford, M. Keller, M. Kloppmann, D. König, F. Leymann, R. Müller, G. Pfau, K. Plösser, R. Rangaswamy, A. Rickayzen, M. Rowley, P. Schmidt, I. Trickovic, A. Yiu, M. Zeller, Web services human task (ws-humantask). version 1.0, http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel4people/WS-HumanTask_v1.pdf (sighted 1 May 2009), June 2007
29. A. Agrawal, M. Amend, M. Das, M. Ford, M. Keller, M. Kloppmann, D. König, F. Leymann, R. Müller, G. Pfau, K. Plösser, R. Rangaswamy, A. Rickayzen, M. Rowley, P. Schmidt, I. Trickovic, A. Yiu, M. Zeller. Ws-bpel extension for people (bpel4people). version 1.0, http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel4people/BPEL4People_v1.pdf (sighted 1 May 2009), 2007

30. R. Agrawal, D. Gunopulos, F. Leymann, Mining process models from workflow logs, ed. by H-J. Schek, F. Saltor, I. Ramos, G. Alonso. In *Proceedings of the 6th International Conference on Extending Database Technology (EDBT'98)*, Lecture Notes in Computer Science, vol. 1377, Valencia, Spain, March 1998 (Springer, Berlin, 1998), pp. 469–484
31. M. Aguilar, A.J.G. Pater, Business process simulation: A fundamental step supporting process centered management, ed. by P. Farrington, H. Nembhard, D. Sturrock, G. Evans. In *Proceedings of the 1999 Winter Simulation Conference (WSC'99)*, Phoenix, AZ, USA, December 1999 (ACM, NY, 1999), pp. 1383–1392
32. L. Aldred, W.M.P. van der Aalst, M. Dumas, A.H.M. ter Hofstede, Communication abstractions for distributed business processes, ed. by J. Krogstie, A. Opdahl, G. Sindre. In *Proceedings of the 19th International Conference on Advanced Information Systems Engineering (CAiSE'07)*, Lecture Notes in Computer Science, vol. 4495, Trondheim, Norway, June 2007 (Springer, Berlin, 2007), pp. 409–423
33. L. Aldred, W.M.P. van der Aalst, M. Dumas, A.H.M. ter Hofstede, Dimensions of coupling in middleware. Concurrency Comput. Pract. Ex. (2009). doi: 10.1002/cpe.1414
34. G. Alonso, D. Agrawal, A. El Abbadi, M. Kamath, G. Gunthor, C. Mohan, Advanced transaction models in workflow contexts, ed. by S.Y.W. Su. In *Proceedings of the 12th International Conference on Data Engineering (ICDE'96)*, New Orleans, LA, USA, February 1996 (IEEE Computer Society, MD, 1996), pp. 574–581
35. A. Alves, A. Arkin, S. Askary, C. Barreto, B. Bloch, F. Curbera, M. Ford, Y. Goland, A. Guízar, N. Kartha, C.K. Liu, R. Khalaf, D. König, M. Marin, V. Mehta, S. Thatte, D. van der Rijn, P. Yendluri, A. Yiu, Web services business process execution language. version 2.0, OASIS standard, http://www.oasis-open.org/specs/#wsbpelv2.0 (sighted 1 May 2009), 2007
36. T. Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, S. Weerawarana, Business process execution language for web services. version 1.1, http://xml.coverpages.org/BPELv11-May052003Final.pdf (sighted 27 April 2009), 2003
37. E. Badouel, P. Darondeau, Theory of regions, ed. by W. Reisig, G. Rozenberg. Lectures on Petri Nets I: Basic Models, Advances in Petri Nets, Lecture Notes in Computer Science, vol. 1491 (Springer, Berlin, 1998), pp. 529–586
38. A. Bahrami, D. Sadowski, S. Bahrami, Enterprise architecture for business process simulation, ed. by D. Medeiros, E. Watson, J. Carson, M. Manivannan. In *Proceedings of the 1998 Winter Simulation Conference (WSC'98)*, Washington, DC, USA, December 1998 (ACM, NY, 1998), pp. 1409–1413
39. J.E. Bardram, I love the system – I just don't use it! ed. by S.C. Payne, W. Prinz In *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work (GROUP'97)*, Phoenix, AZ, USA, November 1997 (ACM, NY, 1997), pp. 251–260
40. L. Baresi, E. Di Nitto (ed.), *Test and Analysis of Web Services* (Springer, Berlin, 2007)
41. P. Barthelmess, J. Wainer, Workflow systems: a few definitions and a few suggestions, ed. by N. Comstock, C. Ellis. In *Proceedings of the ACM Conference on Organizational Computing Systems (COOCS'95)*, Milpitas, CA, USA, 1995 (ACM, NY, 1995), pp. 138–147
42. J. Becker, P. Delfmann, R. Knackstedt, Adaptive reference modeling: Integrating configurative and generic adaptation techniques for information models, ed. by J. Becker, P. Delfmann. In *Reference Modeling – Efficient Information Systems Design Through Reuse of Information Models, Proceedings of the Reference Modeling Conference (RM'06)*, Passau, Germany, February 2007 (Springer, Berlin, 2007), pp. 27–58
43. J. Becker, M. Kugeler, M. Rosemann (ed.), *Process Management. A Guide for the Design of Business Processes* (Springer, Berlin, 2003)
44. J. Becker, R. Schütte, *Handelsinformationssysteme*, (in German) 2nd edn. (Redline Wirtschaft, Frankfurt am Main, Germany, 2004)
45. P. Berens, in *The FLOWer Case Handling Approach: Beyond Workflow Management*, chapter 15, ed. by M. Dumas et al. Process-Aware Information Systems: Bridging People and Software Through Process Technology (Wiley-Interscience, NY, 2005), pp. 363–395

46. P. Bichler, G. Preuner, M. Schrefl, Workflow transparency, ed. by A. Olivé, J.A. Pastor. In *Proceedings of the 9th International Conference on Advanced Information Systems Engineering (CAiSE'97)*, Lecture Notes in Computer Science, vol. 1250, Barcelona, Spain, June 1997 (Springer, Berlin, 1997), pp. 423–436
47. A. Borgida, T. Murata, Tolerating exceptions in workflows: A unified framework for data and processes, ed. by D. Georgakopoulos, W. Prinz, A.L. Wolf. In *Proceedings of the International Joint Conference on Work Activities Coordination and Collaboration (WACC'99)*, San Francisco, CA, USA, February 1999 (ACM, NY, 1999), pp. 59–68
48. M. Brambilla, S. Ceri, S. Comai, C. Tziviskou, Exception handling in workflow-driven web applications, ed. by A. Ellis, T. Hagino. In *Proceedings of the 14th International Conference on World Wide Web (WWW2005)*, Chiba, Japan, May 2005 (ACM, NY, 2005), pp. 170–179
49. E. Bredenhoff, R. Schuring, R. Caljouw, Zorg in focus, op zoek naar routine in het zorgproces. (in Dutch). Medisch Contact **59**(34), 1304–1308 (2004)
50. F. van Breugel, M. Koshkina, Models and verification of BPEL, http://www.cse.yorku.ca/~franck/research/drafts/tutorial.pdf (sighted 1 May 2009), 2006
51. A. Brogi, R. Popescu, From BPEL processes to YAWL workflows, ed. by M. Bravetti, M. Núñez, G. Zavattaro. In *Proceedings of the 3rd International Workshop on Web Services and Formal Methods (WS-FM'06)*, Lecture Notes in Computer Science, vol. 4148, Vienna, Austria, 2006 (Springer, Berlin, 2006), pp. 107–122
52. C. Bussler, *B2B Integration: Concepts and Architecture* (Springer, Berlin, 2003)
53. J.A. Buzacott, Commonalities in reengineered business processes: Models and issues. Manag. Sci. **42**(5), 768–782 (1996)
54. F. Casati, A discussion on approaches to handling exceptions in workflows. ACM SIGGROUP Bull. **20**(3), 3–4 (1999)
55. F. Casati, S. Ceri, S. Paraboschi, G. Pozzi, Specification and implementation of exceptions in workflow management systems. ACM Trans. Database Syst. **24**(3), 405–451 (1999)
56. F. Casati, S. Ilnicki, L. Jin, V. Krishnamoorthy, M-C. Shan, Adaptive and dynamic composition in eFlow, ed. by B. Wangler, L. Bergman. In *Proceedings of the 12th International Conference on Advanced Information Systems Engineering (CAiSE 2000)*, Lecture Notes in Computer Science, vol. 1789, Stockholm, Sweden, 2000 (Springer, Berlin, 2000), pp. 13–31
57. D. Chiu, Q. Li, K. Karlapalem, A logical framework for exception handling in ADOME workflow management system, ed. by B. Wangler, L. Bergman. In *Proceedings of the 12th International Conference on Advanced Information Systems Engineering (CAiSE 2000)*, Lecture Notes in Computer Science, vol. 1789, Stockholm, Sweden, 2000 (Springer, Berlin, 2000), pp. 110–125
58. D.K.W. Chiu, Q. Li, K. Karlapalem, ADOME-WFMS: Towards cooperative handling of workflow exceptions, ed. by A. Romanovsky, C. Dony, J. Lindskov Knudsen, A. Tripathi. In *Advances in Exception Handling Techniques*, Lecture Notes in Computer Science, vol. 2022 (Springer, Berlin, 2001), pp. 271–288
59. R. Chokshi, Decision support for call center management using simulation, ed. by P. Farrington, H. Nembhard, D. Sturrock, G. Evans. In *Proceedings of the 1999 Winter Simulation Conference (WSC'99)*, Phoenix, AZ, USA, December 1999 (ACM, NY, 1999), pp. 1634–1639
60. E.M. Clarke, O. Grumberg, D.A. Peled, *Model Checking* (MIT, Cambridge, MA, 1999)
61. B. Clevé, *Film Production Management*, 3rd edn. (Focal Press, Burlington, MA, 2006)
62. P. Compton, B. Jansen, Knowledge in context: A strategy for expert system maintenance, ed. by J. Siekmann. In *Proceedings of the 2nd Australian Joint Artificial Intelligence Conference (AI'88)*, Lecture Notes in Artificial Intelligence, vol. 406, Adelaide, Australia, November 1988 (Springer, Berlin, 1988), pp. 292–306
63. P. Compton, P. Preston, B. Kang, The use of simulated experts in evaluating knowledge acquisition. In *Proceedings of the 9th Banff Knowledge Acquisition for Knowledge Based Systems Workshop*, Banff, Canada, February 1995
64. J. Cortadella, *Petrify: a tutorial for the designer of asychronous circuits*, Universitat Politécnica de Catalunya, Barcelona, Spain, http://www.lsi.upc.es/petrify (sighted 20 May 2007), December 1998

65. J. Cortadella, M. Kishinevsky, L. Lavagno, A. Yakovlev, Deriving Petri nets from finite transition systems. IEEE Trans. Comput. **47**(8), 859–882 (1998)
66. COSA BPM product description, http://www.cosa-bpm.com/project/docs/COSA_BPM_5_Productdescription_eng.pdf (sighted 13 March 2007), 2005
67. M. Cumberlidge, *Business Process Management with JBoss jBPM: A Practical Guide for Business Analysts* (PACKT Publishing, Birmingham, UK, 2007)
68. F. Curbera, M.J. Duftler, R. Khalaf, W. Nagy, N. Mukhi, S. Weerawarana, Unraveling the web services web: An introduction to SOAP, WSDL, and UDDI. IEEE Internet Comput. **6**(2), 86–93 (2002)
69. F. Curbera, Y. Goland, J. Klein, F. Leymann, D. Roller, S. Thatte, S. Weerawarana, Business process execution language for web services, version 1.0. Standards proposal by BEA Systems, International Business Machines Corporation, and Microsoft Corporation (2002)
70. T. Curran, G. Keller, A. Ladd, SAP R/3 Business Blueprint: Understanding the Business Process Reference Model, in *Enterprise Resource Planning Series* (Prentice Hall PTR, NJ, 1997)
71. K. Czarnecki, M. Antkiewicz, Mapping features to models: A template approach based on superimposed variants, ed. by R. Glück, M.R. Lowry. In *Proceedings of the 4th International Conference on Generative Programming and Component Engineering (GPCE'05)*, Lecture Notes in Computer Science, vol. 3676, Tallinn, Estonia, September 2005 (Springer, Berlin, 2005), pp. 422–437
72. P. Darondeau, Unbounded Petri net synthesis, ed. by J. Desel, W. Reisig, G. Rozenberg. In *Lectures on Concurrency and Petri Nets, Advances in Petri Nets*, Lecture Notes in Computer Science, vol. 3098, Eichstätt, Germany, 2003 (Springer, Berlin, 2003), pp. 413–428
73. G. Decker, A. Grosskopf, A. Barros, A graphical notation for modeling complex events in business processes, in *Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC'07)*, Annapolis, MD, USA, October 2007 (IEEE Computer Society, MD, 2007), pp. 27–36
74. G. Decker, J. Mendling, Instantiation semantics for process models, ed. by M. Dumas, M. Reichert, M.-C. Shan. In *Proceedings of the 6th International Conference on Business Process Management (BPM'08)*, Lecture Notes in Computer Science, vol. 5240, Milan, Italy, September 2008 (Springer, Berlin, 2008), pp. 164–179
75. J. Dehnert, *A Methodology for Workflow Modelling: from Business Process Modelling towards Sound Workflow Specification*, PhD thesis, Technische Universität Berlin, Berlin, Germany, 2003
76. W. Derks, J. Dehnert, P. Grefen, W. Jonker, Customized atomicity specification for transactional workflows, ed. by H. Lu, S. Spaccapietra. In *Proceedings of the 3rd International Symposium on Cooperative Database Systems and Applications (CODAS'2001)*, Beijing, China, April 2001 (IEEE Computer Society, MD, 2001), pp. 155–164
77. J. Desel, J. Esparza, *Free Choice Petri Nets*. Cambridge Tracts in Theoretical Computer Science, vol. 40 (Cambridge University Press, Cambridge, UK, 1995)
78. R. Dijkman, M. Dumas, C. Ouyang, Formal semantics and analysis of BPMN process models. Inform. Software Tech. **50**(12), 1281–1294 (2008)
79. B.F. van Dongen, W.M.P. van der Aalst, A meta model for process mining data, ed. by M. Missikoff, A. De Nicola. In *Proceedings of the CAiSE'05 Workshops (EMOI-INTEROP Workshop)*, CEUR Workshop Proceedings, vol. 160, Porto, Portugal, June 2005, pp. 309–320, CEUR-WS.org
80. C. Dufourd, A. Finkel, P. Schnoebelen, Reset nets between decidability and undecidability, ed. by K. Larsen, S. Skyum, G. Winskel. In *Proceedings of the 25th International Colloquium on Automata, Languages and Programming (ICALP'98)*, Lecture Notes in Computer Science, vol. 1443, Aalborg, Denmark, July 1998 (Springer, Berlin, 1998), pp. 103–115
81. C. Dufourd, P. Jančar, P. Schnoebelen, Boundedness of reset P/T nets, ed. by J. Wiedermann, P. van Emde Boas, M. Nielsen. In *Proceedings of the 26th International Colloquium on Automata, Languages and Programming (ICALP'99)*, Lecture Notes in Computer Science, vol. 1644, Prague, Czech Republic, July 1999 (Springer, Berlin, 1999), pp. 301–310

82. M. Dumas, W.M.P. van der Aalst, A.H.M. ter Hofstede (ed.), *Process-Aware Information Systems: Bridging People and Software through Process Technology* (Wiley-Interscience, NY, 2005)
83. M. Dumas, A. Grosskopf, T. Hettel, M.T. Wynn, Semantics of standard process models with OR-joins, ed. by R. Meersman, Z. Tari. In *Proceedings of the 15th International Conference on Cooperative Information Systems (CoopIS'07)*, Lecture Notes in Computer Science, vol. 4803, Vilamoura, Portugal, November 2007 (Springer, Berlin, 2007), pp. 41–58
84. M. Dumas, A.H.M. ter Hofstede, UML activity diagrams as a workflow specification language, ed. by M. Gogolla, C. Kobryn. In *Proceedings of the 4th International Conference on the Unified Modeling Language (UML 2001)*, Lecture Notes in Computer Science, vol. 2185, Toronto, Canada, October 2001 (Springer, Berlin, 2001), pp. 76–90
85. Eastman Software, *RouteBuilder Tool User's Guide* (Eastman Software Inc, Billerica, MA, 1998)
86. J. Eder, W. Liebhart, The workflow activity model (WAMO), ed. by S. Laufmann, S. Spaccapietra, T. Yokoi. In *Proceedings of the 3rd International Conference on Cooperative Information Systems (CoopIS'95)*, Vienna, Austria, May 1995, pp. 87–98
87. J. Eder, W. Liebhart, Workflow recovery, in *Proceedings of the 1st IFCIS International Conference on Cooperative Information Systems (CoopIS'96)*, Brussels, Belgium, June 1996 (IEEE Computer Society, MD, 1996), pp. 124–134
88. A. Ehrenfeucht, G. Rozenberg, Partial (set) 2-structures - part 1 and part 2. Acta Inform. **27**(4), 315–368 (1989)
89. C.A. Ellis, K. Keddara, G. Rozenberg, Dynamic change within workflow systems, ed. by N. Comstock, C. Ellis, R. Kling, J. Mylopoulos, S. Kaplan. In *Proceedings of the Conference on Organizational Computing Systems*, Milpitas, CA, USA, August 1995 (ACM, NY, 1995), pp. 10–21
90. A. Elmagarmid (ed.), *Database Transaction Models for Advanced Applications* (Morgan Kaufmann, CA, 1992)
91. P. Eugster, P. Felber, R. Guerraoui, A. Kermarrec, The many faces of publish/subscribe. ACM Comput. Surv. **35**(2), 114–131 (2003)
92. P. Fettke, P. Loos, Classification of reference models - a methodology and its application. Inform. Syst. E. Bus. Manag. **1**(1), 35–53 (2003)
93. P. Fettke, P. Loos, J. Zwicker, Business process reference models: Survey and classification, ed. by C. Bussler, A. Haller et al. In *Business Process Management Workshops – BPM 2005 International Workshops, BPI, BPD, ENEI, BPRM, WSCOBPM, BPS. Revised Selected Papers*, Lecture Notes in Computer Science, vol. 3812, Nancy, France, September 2005 (Springer, Berlin, 2005), pp. 469–483
94. R.T. Fielding, *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, CA, USA, 2000
95. A. Finkel, J.-F. Raskin, M. Samuelides, L. Van Begin, Monotonic extensions of petri nets: Forward and backward search revisited. Electron. Notes Theor. Comput. Sci. **68**(6), 1–22 (2002)
96. A. Finkel, P. Schnoebelen, Well-structured transition systems everywhere! Theor. Comput. Sci. **256**(1–2), 63–92 (2001)
97. L. Fischer (ed.), *Workflow Handbook 2004* (Future Strategies Inc, Lighthouse Point, FL, 2004)
98. M. Fowler, *Patterns of Enterprise Application Architecture* (Addison-Wesley, Boston, MA, 2003)
99. B.R. Gaines, Induction and visualisation of rules with exceptions, ed. by J. Boose, B. Gaines. In *Proceedings of the 6th Banff AAAI Knowledge Acqusition of Knowledge Based Systens Workshop*, Banff, Canada, October 1991, pp. 7.1–7.17
100. E. Gamma, R. Helm, R. Johnson, J. Vlissides, In *Design Patterns: Elements of Reusable Object-Oriented Software*. Professional Computing Series (Addison-Wesley, Boston, MA, 1995)
101. D. Georgakopoulos, M.F. Hornick, A.P. Sheth, An overview of workflow management: From process modeling to workflow automation infrastructure. Distr. Parallel Database **3**(2), 119–153 (1995)