Protos 7.0: Simulation Made Accessible

Eric Verbeek[1], Maarte van Hattem[2], Hajo Reijers[1], and Wendy de Munk[2]

[1] Department of Technology Management, Eindhoven University of Technology,
P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands
{h.m.w.verbeek, h.a.reijers}@tm.tue.nl
[2] Pallas Athena BV,
P.O. Box 747, NL-7300 AS Apeldoorn, The Netherlands
service@pallas-athena.com

Abstract. Many consider simulation to be a highly specialist activity: it is difficult to undertake and is even more difficult to understand its outcomes. The new version of the business process modeling tool Protos attempts to more closely integrate modeling and simulation facilities into one tool. The assumed benefit is that business professionals may more easily undertake simulation experiments when they are enabled with the same tool to extend their existing process models to carry out simulation experiments. This paper explains how the existing engine of the Petri-net based tool ExSpect is integrated into Protos 7.0. It also shows the extended user interface of Protos and the simulation reports it generates.

1 Introduction

In the early 90s, it was no less than a revolution when companies started focusing their attention on the performance of entire business processes. In the 21st century, "process thinking" has become mainstream organizational practice [11]. The pursuit of specifically supporting business processes with all kinds of methods, techniques, and systems has become known as Business Process Management (BPM) [2].

For any BPM effort, which includes designing, enacting, controlling, or analyzing a business process, it is important to have a conception of the business process in question. Various approaches exist to model such business processes, but particularly graphical approaches such as the UML and IDEF have become prevalent. Petri nets too have become quite popular in this respect, resulting in various proposed Petri net classes (see e.g. [8]). The influence of Petri nets on other modeling languages is substantial; consider for example how they inspired UML's Activity Diagrams. Next to widespread academic and industrial use of Petri nets for the purpose of business process modeling, various commercial vendors made Petri nets the backbone of their process-aware information systems. Consider, for example, the COSA Workflow Management System and the Dynamic Enterprise Modeler of the SSA ERP_{LN} System (formerly known as BAAN).

G. Ciardo and P. Darondeau (Eds.): ICATPN 2005, LNCS 3536, pp. 465–474, 2005.

The BPM modeling and analysis tool that we will focus on in this paper is Protos from the company Pallas Athena. Current versions of Protos are in use by thousands of organizations in more than 25 countries. In The Netherlands alone, more than half of all municipalities use Protos for the specification of their in-house business processes. Protos is perfectly suitable to model well-defined Petri net structures. Nevertheless, it also permits freehand specifications of business processes without formal semantics, e.g. to support initial and conceptual modeling. In an earlier publication [1] we reported on the use of Protos in the context of the Petri-net based ExSpect tool. We showed how a specific set of Protos models could be automatically translated to be enacted and analyzed with the tool ExSpect (for an introduction, see [7]). Due to the formal Petri net semantics of Protos models, translations to various other process-based systems are feasible as well, e.g. to the workflow management system COSA and the workflow analyzer Woflan [12].

The main use of Protos is to define models of business processes as a step towards either the implementation of quality systems, the redesign of a business process, communication enhancement between process stakeholders, or the implementation of workflow management systems. Until now, Protos only allowed for basic analyses of the models. For example, it is possible to view a Protos process model from different perspectives, i.e. the data, user, or control logic perspective. Considering the main use of Protos, it can perhaps be imagined that a more quantitative analysis of a business process may be valuable. For example, ahead of the implementation of a workflow management system, a simulation of a process assuming such a system to be in place could give insight into the attractiveness of pursuing an implementation project. Similarly, alternative redesign scenarios of a business process could be compared in terms of their ability to meet due dates, execution cost, etc. In short, many of the advantages of simulation seem to be applicable in the realm of business processes, just as they are in the design of more tangible artifacts such as airplanes and cars.

Taking into account the popularity of process modeling, it may come as a surprise that simulation is hardly applied by the same business professionals that use tools such as Protos for modeling. Simulation is considered by many as a highly specialist activity, which is difficult to undertake, while it is even more difficult to understand its outcomes. Protos 7.0 is an attempt to more closely integrate modeling and simulation facilities into one tool. The assumed benefit is that business professionals may more easily undertake simulation experiments when they find out that they could use the same model and the same tool for this purpose. By restricting the simulation settings and the development of a simple user interface, it is expected that the parameterization of a business model is understandable and executable by most business professionals. By creating a standardized simulation report, it is expected that the analysis of a simulation becomes easier too. The main technological means to accomplish the integration is to incorporate the engine of the Petri-net based package ExSpect into the Protos software architecture.

This paper is structured as follows. In section 2, the main architecture of Protos 7.0 is explained. Section 3 presents a short tour through the user interface of the tool, illustrated with an example of the application of Protos 7.0. This paper ends with a conclusion that reflects on the envisioned use of Protos and future work.

2 Architecture

In a bird's eye view, Protos 7.0 embeds the ExSpect COM-server [1] (which we simply call the server from now on) and communicates with the server by means of two XML [4] formatted streams. The first formatted stream is used to communicate the process modeled in Protos to the server; the second stream is used to communicate the simulation report back to Protos. For ease of reference, we will call these streams the process stream and the report stream. Because the formats of both streams give a good overview of the simulation features offered by Protos 7.0, this section will go into details about them.

To map the process stream onto an ExSpect stream, an XSLT [5] file is used. One advantage of doing so is that by replacing this XSLT file we can also feed the process stream to a different simulation server. For example, we can use an XSLT file that maps the process stream onto a CPN Tools [10] stream, and use CPN Tools to obtain a simulation report.

2.1 Process Stream

The following figures are XMLSpy content model views [3], which visualize the XML schema [6] that underlies the process stream. The root element of the

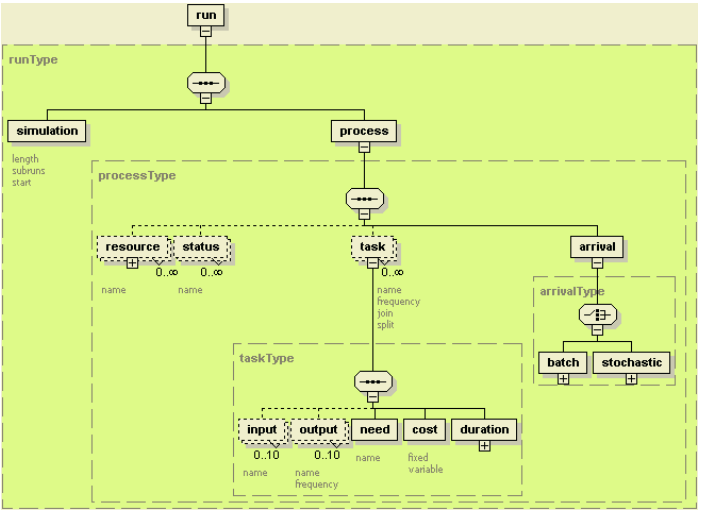

Fig. 1. Format of a process

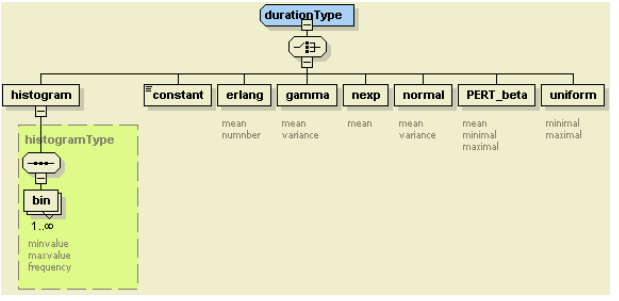

Fig. 2. Format of duration (and histogram)

schema is called *run*, for which Figure 1 shows a high level overview. From this figure, we learn for example, (i) that a *process* element contains (in the given order) a set of *resource* elements, a set of *status* elements, a set of *task* elements, and an *arrival* element, and (ii) that an *arrival* element contains either a *batch* element or a *stochastic* element. For some elements shown by Figure 1 the contained elements are not shown (which would take too much space), which is indicated by a plus sign underneath the element. For these elements, namely *resource*, *batch*, *duration*, and *stochastic*, Figure 2 shows the contained elements, where elements *resource* and *batch* have type *histogramType* and elements *duration* and *stochastic* have type *durationType*.

The *simulation* element prescribes the length of one subrun (*length*), the number of subruns used for measurements (*subruns*), and the number of subruns used to obtain a reasonable starting state (and for which the measurements are discarded) (*start*). Thus, the length of the entire simulation run equals (*start*+*subruns*)**length*. A *resource* element contains only the name of a particular resource class (or role) (*name*) and a histogram that describes the number of available resources of this class at a specific time. A *status* element contains only its name, and basically corresponds to a place in a Petri net. A *task* element basically corresponds to a transition in a Petri net and contains up to 10 names of input statuses (*input*), 10 names of output statuses (*output*), the name of the resource which should perform the task (*need*), the fixed and variable cost of performing this task (*cost*), and a function for determining the time it takes to perform it (*duration*). Furthermore, the task contains its name (*name*), its frequency (*frequency*), and its join and split behavior (*join*, *split*). A task frequency determines the relative weight of a task and is relevant only if it is involved in a choice. If two tasks with frequencies $f1$ and $f2$ share input places, and if both tasks can be chosen, then the probabilities the tasks are chosen are $f1/(f1 + f2)$ and $f2/(f1 + f2)$. Please note that output statuses contain also frequencies, which determine the chosen output status if that tasks split behavior is XOR. A task's join and split behavior can either be AND (all input/output statuses) or XOR (exactly one input/output status). A *batch* element results in a batch-driven arrival process, whereas a stochastic element results in a case-based arrival process with stochastic inter-arrival times.

2.2 Report Stream

Figure 3 visualizes the XML schema that underlies the report stream. The report contains the following items:

- the utilization rate of a resource class (*resource/utilization*), that is, the part of the time available resources of this class were actually working on some task;
- the wait time of a status (*status/waitTime*), that is, the average time a case has to wait for synchronization in this status;
- the combined wait and queue time of a status (*status/waitQueueTime*), that is, the average time a case spends in the corresponding status;
- the cost of a task (*task/cost*);
- the queue time of a task (*task/queueTime*), that is, the average time a case has to wait for a resource to become available for it;
- the work time of a task (*task/workTime*), that is, the average time it takes to perform this task for any case;
- the sojourn time of cases (*time*), that is, the average time a case spends in the entire process;
- the cost of cases (*cost*), that is, the average cost per case for the entire process;
- the work time of cases (*workTime*), that is, the average work time per case for the entire process.

As Figure 3 shows, the report contains for each item an estimated average (*mean*), a 90% confidence interval (*lo90* up to *hi90*) and a 99% confidence interval (*lo99* up to *hi99*).

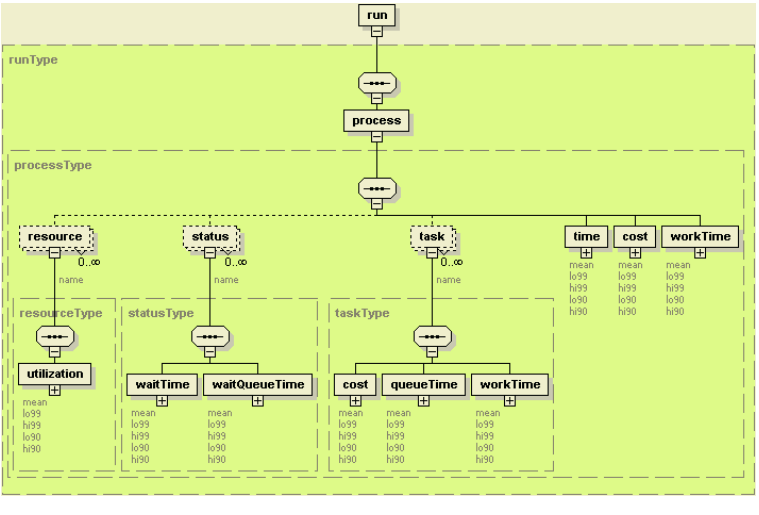

Fig. 3. Format of a report

3 User Interface

The User Interface for the simulation feature in Protos 7.0 includes native forms to parameterize the process stream, and Microsoft Excel to inspect the (converted) report stream.

3.1 Parameterizing the Process Stream

Figure 4 shows a simplified complaint-handling process of an actual Dutch inspection agency, modeled using Protos. After a complaint has been received, we first check whether the complaint is for the inspection agency or not. If not, the complaint will be referred. The inspection agency first stores the relevant data on both the complainant and the complaint, and then determines the procedure to be followed etc.

If we take a closer look at the properties of one activity, for instance *Determine procedure*, we discover the role occupied with the execution of this ac-

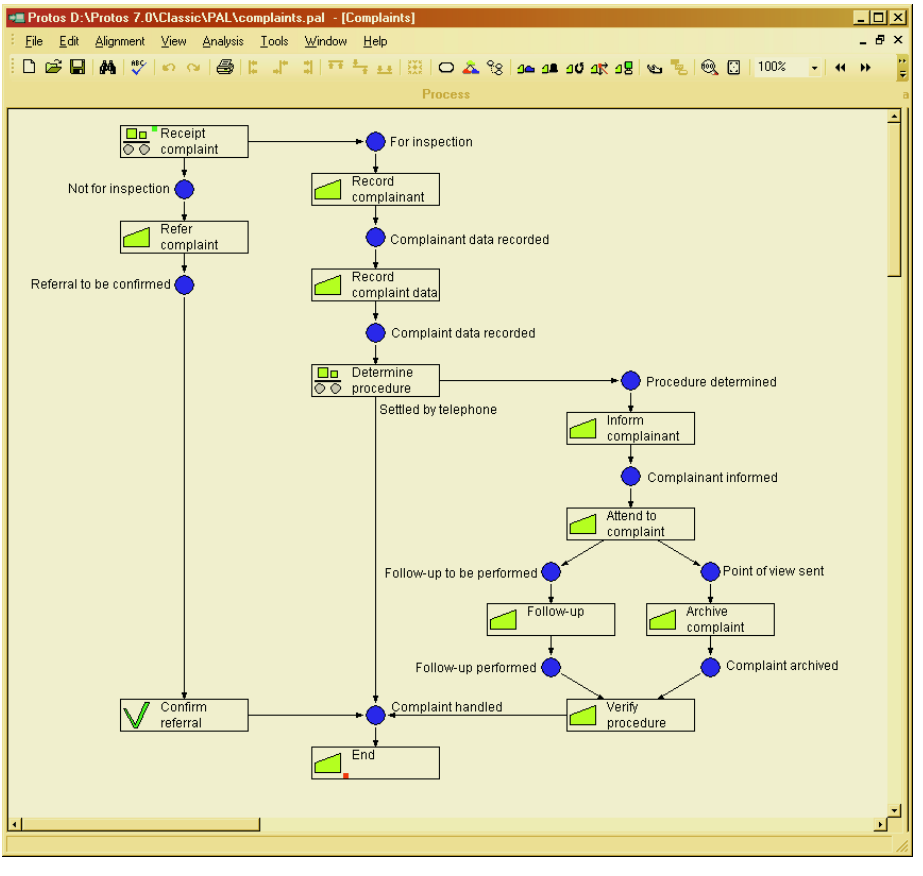

Fig. 4. An example process modeled in Protos

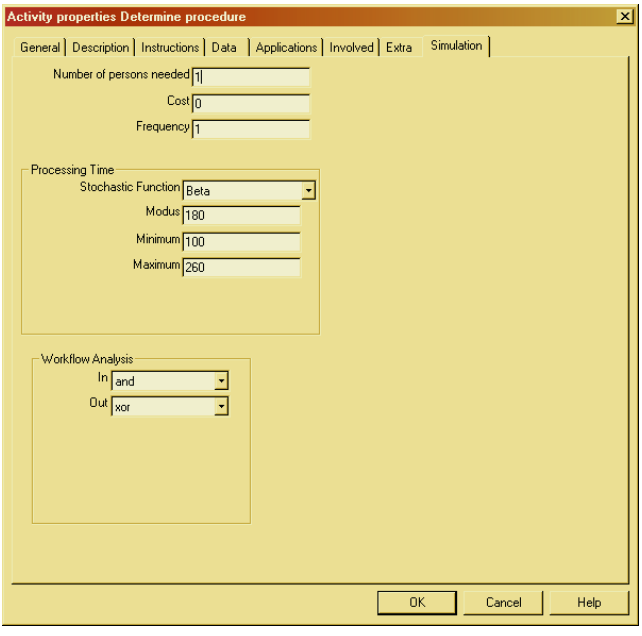

Fig. 5. Setting simulation parameters for activity determine procedure

Table 1. Simulation parameters on non-activity levels

Object	**Parameters**	
Role	Number of persons	
	Cost	
Connection	Frequency	
Trigger	Frequency	
Proces	Number of Sub-runs	
	Length of Sub-run	
	Number of Start-runs	
	Arrival Time of Cases	Stochastic Function

tivity. We also see the tab to set the simulation parameters; this is shown in Figure 5.

Figure 5 shows the simulation parameters we can set for the activity *Determine procedure*. An activity corresponds to a *task* element in the process stream, and every parameter corresponds to some element from that task element: a number of resources corresponds to a *resource* element (a one-bin histogram), a resource class corresponds to a *need* element, a fixed cost corresponds to the *fixed* attribute of a *cost* element, and a frequency corresponds to the *frequency* attribute.

At the top, we can set the number of resources needed to perform this activity (which resource class is needed is set on the *General* tab), the fixed cost of performing this activity, and its frequency.

At the middle, we can set a stochastic function to determine the time needed to perform this activity for the case at hand (*duration* element) together with its parameters. In this case, the activity *Determine procedure* uses a PERT Beta [9] function with an optimistic time of 100 units (*minimum*), a pessimistic time of 260 units (*maximum*), and a most likely time of 180 units (*modus*).

At the bottom, we can set the join and split behavior of the activity (*join* and *split* attribute). In this case, the activity *Determine procedure* synchronizes (*and*) cases on all input places, and selects exactly one output state (*xor*) to forward the case to. Except for the parameters on the activity level, parameters on four other levels can be set: on the process level, on the resource level, on the arc level (on output arcs, frequencies can be set), and on the trigger level. Table 1 shows these levels with their parameters.

3.2 Inspecting the Report

Figure 6 shows the final report (using Microsoft Excel). For the sake of completeness, we mention that we changed the format of the data cells (two

Microsoft Excel - complaints.xls

		Activities												
		Queue Time					Work Time					Cost		
		Mean	lo90	hi90	lo99	hi99	Mean	lo90	hi90	lo99	hi99	Mean	lo90	hi90
12	Receipt complaint	0.41	0.15	0.66	-0.06	0.88	1.82	1.30	2.35	0.84	2.80	1.82	1.30	2.35
13	Attend to complaint	43.58	-4.01	91.17	-44.55	131.71	280.10	203.60	356.60	138.43	421.77	280.10	203.60	356.60
14	Refer complaint	0.20	0.02	0.37	-0.13	0.52	1.77	1.27	2.27	0.84	2.70	1.77	1.27	2.27
15	Follow-up	36.38	-11.74	84.49	-52.73	125.48	110.81	80.20	141.43	54.12	167.51	110.81	80.20	141.43
16	Archive complaint	0.48	0.12	0.84	-0.19	1.14	4.40	3.21	5.59	2.19	6.60	4.40	3.21	5.59
17	Record complainant data	86.91	10.45	163.37	-54.68	228.50	12.69	9.24	16.14	6.30	19.07	12.69	9.24	16.14
18	Confirm referral	0.14	-0.03	0.32	-0.18	0.47	5.11	3.66	6.56	2.43	7.79	5.11	3.66	6.56
19	Verify procedure	8.27	2.77	13.77	-1.91	18.45	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	End	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	Record complaint data	26.99	7.32	46.66	-9.44	63.42	14.56	10.62	18.50	7.27	21.85	14.56	10.62	18.50
22	Determine procedure	16.72	-1.40	34.84	-16.83	50.28	155.70	113.44	197.96	77.44	233.96	155.70	113.44	197.96
23	Inform complainant	0.25	0.04	0.45	-0.13	0.62	4.67	3.40	5.94	2.31	7.02	4.67	3.40	5.94

		Status									
		wait time					wait queue time				
		Mean	lo90	hi90	lo99	hi99	Mean	lo90	hi90	lo99	hi99
28	For inspection	0.00	0.00	0.00	0.00	0.00	86.91	10.45	163.37	-54.68	228.50
29	Not for inspection	0.00	0.00	0.00	0.00	0.00	0.20	0.02	0.37	-0.13	0.52
30	Follow-up to be performed	0.00	0.00	0.00	0.00	0.00	36.38	-11.74	84.49	-52.73	125.48
31	Point of view sent	0.00	0.00	0.00	0.00	0.00	0.48	0.12	0.84	-0.19	1.14
32	Referral to be confirmed	0.00	0.00	0.00	0.00	0.00	0.14	-0.03	0.32	-0.18	0.47
33	Follow-up performed	0.00	0.00	0.00	0.00	0.00	8.27	2.77	13.77	-1.91	18.45
34	Complaint archived	144.42	82.78	206.05	30.28	258.55	152.69	89.68	215.69	36.01	269.36
35	Complaint handled	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
36	Complainant data recorded	0.00	0.00	0.00	0.00	0.00	26.99	7.32	46.66	-9.44	63.42
37	Complaint data recorded	0.00	0.00	0.00	0.00	0.00	16.72	-1.40	34.84	-16.83	50.28
38	Procedure determined	0.00	0.00	0.00	0.00	0.00	0.25	0.04	0.45	-0.13	0.62
39	Complainant informed	0.00	0.00	0.00	0.00	0.00	43.58	-4.01	91.17	-44.55	131.71

	Total													
	Time					Work Time					Cost			
	Mean	lo90	hi90	lo99	hi99	Mean	lo90	hi90	lo99	hi99	Mean	lo90	hi90	lo99
44	335.6722259	179.31	492.04	46.11	625.24	226.82	148.69	304.94	82.15	371.49	226.82	148.69	304.94	82.15

Fig. 6. Report for the example process

digits following the decimal point) and reduced the column width of the data columns.

At the moment, a report can only be inspected using this spreadsheet format. At a later stage, we plan to incorporate the possibility for inspection (and visualization) into the Protos tool.

A report like this reveals a lot of information. For example, from Figure 6 we can conclude that the wait time at the *Complaint archived* status is excessive and needs to be reduced. Furthermore, such a report could be used for validating the process modeled by comparing the results against measurements taken from the real-world situation.

4 Conclusion

In this paper, we have presented Protos 7.0, which extends the popular business process modeling tool Protos with simulation capabilities. We showed how the technical integration of the engine of the Petri-net based ExSpect tool into Protos 7.0 was established. Using a simplified case from practice, we illustrated how Protos models can be extended with various parameter settings so that the simulation of such models becomes feasible. Finally, we showed what the output of a simulation looks like in Protos 7.0 and how such output may be used to improve the design of the process.

As Protos 7.0 has been released only recently, we have no figures on industrial usage of the simulation facility. A beta version of the tool has been introduced recently in a course of the bachelor education program on industrial engineering at Eindhoven University of Technology. This has led to various improvements of the tool, but, more importantly, seems to confirm the relative ease of creating simulation models by non-simulation experts. Considering this and the widespread use of earlier Protos versions, we expect business professionals to get increasingly involved in simulation experiments. Nevertheless, the gathering of accurate data needed as input for a simulation still needs to be performed by professionals.

In upcoming versions of the tool, we plan to improve the tool in several ways. First, we plan to enable the process designer to specify histograms for certain simulation parameters, for example, for resource availability, or for interarrival times. The format of the process stream and the simulation engine already provide support for these histograms, but the user interface does not yet. Second, we plan to integrate the report into the tool. The current version of the tool relies on Microsoft Excel for displaying the report results, while we could display these results in the tool itself. For example, wait times could be displayed near the corresponding status and queue times near to the corresponding activity. Excessive queue and/or wait times and ditto utilization rates, which are likely to correspond to bottlenecks, can even be visualized using, for example, a coloring scheme.

References

1. W.M.P. van der Aalst, P. de Crom, R. Goverde, K.M. van Hee, W. Hofman, H. Reijers, and R.A. van der Toorn. ExSpect 6.4: An executable specification tool for hierarchical colored Petri nets. In M. Nielsen and D. Simpson, editors, *Application and Theory of Petri Nets 2000*, volume 1825 of *Lecture Notes in Computer Science*, pages 455–464. Springer, Berlin, Germany, 2000.
2. W.M.P. van der Aalst, A.H.M. ter Hofstede, and M. Weske. Business process management: A survey. In W.M.P. van der Aalst, A.H.M. ter Hofstede, and M. Weske, editors, *1st International Conference on Business Process Management (BPM 2003)*, volume 2678 of *Lecture Notes in Computer Science*, pages 1–12, Eindhoven, The Netherlands, June 2003. Springer, Berlin, Germany.
3. Altova. XMLSpy online manual: Content model view. http://link.xmlspy.com/manual2005/xmlspy/spyprofessional/contentmodelview.htm, last visited on November 2, 2004.
4. T. Bray, J. Paoli, C.M. Sperberg-McQueen, and E. Maler. eXtensible Markup Language (XML) 1.0 (second edition). http://www.w3.org/TR/REC-xml, 2000.
5. J. Clark. XSL Transformations (XSLT) version 1.0. http://www.w3.org/TR/1999/REC-xslt-19991116, 1999.
6. D.C. Fallside and P. Walmsley. XML Schema part 0, primer second edition. http://www.w3.org/TR/2004/REC-xmlschema-0-20041028, 2004.
7. K.M. van Hee, L.J. Somers, and M. Voorhoeve. Executable specifications for distributed information systems. In E.D. Falkenberg and P. Lindgreen, editors, *Proceedings of the IFIP TC 8 / WG 8.1 Working Conference on Information System Concepts: An In-depth Analysis*, pages 139–156, Namur, Belgium, 1989. Elsevier Science Publishers, Amsterdam.
8. G.K. Janssens, J. Verelst, and B. Weyn. Techniques for modeling workflows and their support of reuse. In W.M.P. van der Aalst, J. Desel, and A. Oberweis, editors, *Business Process Management: Models, Techniques, and Empirical Studies*, volume 1806 of *Lecture Notes in Computer Science*, pages 1–15. Springer, Berlin, Germany, 2000.
9. J.J. Moder and C.R. Phlips. *Project Management with CPM and PERT Second Edition.* Van Nostrand Reinhold, New York, 1970.
10. A.V. Ratzer, L. Wells, H.M. Lassen, M. Laursen, J.F. Qvortrup, M.S. Stissing, M. Westergaard, S. Christensen, and K. Jensen. CPN tools for editing, simulating, and analysing coloured Petri nets. In W.M.P. van der Aalst and E. Best, editors, *24th International Conference on Application and Theory of Petri Nets (ICATPN 2003)*, volume 2679 of *Lecture Notes in Computer Science*, pages 450–462, Eindhoven, The Netherlands, June 2003. Springer, Berlin, Germany.
11. A. Sharp and P. McDermott. *Workflow Modeling: Tools for Process Improvement and Application Development.* Artech House Publishers, Boston, 2001.
12. H.M.W. Verbeek, T. Basten, and W.M.P. van der Aalst. Diagnozing workflow processes using Woflan. *The Computer Journal*, 44(4):246–279, 2001.