Dynamic Process Enactment, Discovery, and Recovery

Walt Scacchi, Institute for Software Research, University of California, Irvine
Irvine, CA 92697-3425 USA, 949-824-4130, Wscacchi@uci.edu

John Noll, Computer Engineering Dept., Santa Clara University
Santa Clara, CA 95053 USA, 408-554-2760, Jnoll@cse.scu.edu
November 2002

Research Goal

Our interest is in understanding three aspects of complex, online knowledge work processes. First is how to provide model-driven process enactment support and event data capture for globally dispersed enterprise processes, resources and users. Second is how to discover process structures and resource usage patterns from emergent and dynamic process enactments. Third is how to recover or repair knowledge work processes that breakdown or fail during enactment. Our goal is to develop and demonstrate concepts, techniques, mechanisms, system architectures, and tools that incorporate these three aspects to enable the subsequent construction of task scheduling and resource allocation/control strategies for coordinating the knowledge work of a distributed complex of people and computing systems.

Problem

The *research area* addressed here is: how to most effectively and efficiently deploy, monitor, learn, and repair the rules, objects, contexts, and teamwork structures that emerge during the enactment of globally distributed knowledge work processes. The *management issue* addressed here is: how to most effectively and efficiently enable flexible process modeling, enactment, reconfiguration and rescheduling of human and computational resources that enables intelligent management response to external change affecting routine, dynamic, or hidden process enactments[1]. Our investigation of these problems is targeted at analysis, design, and prototyping of software system mechanisms, data representations, and system architectures in an iterative and incremental manner, so as to ensure the production and delivery of research results.

Anticipated Results, Deliverables and Transition

We expect to develop and demonstrate concepts, techniques, mechanisms, system architectures and tools that enable the modeling, enactment, discovery, and recovery of complex, online knowledge work processes. The *concepts* will help specify the requirements and design of the tools, techniques, and information infrastructure mechanisms needed to acquire, represent, enact and repair models of dynamic, online knowledge work processes. The *techniques and system architectures* will provide the guiding heuristics and rules of application that coordinate and

[1] In our view, a "process" denotes a set or class of workflows. A computer-based "workflow" specification is an enactable instance of a process. Both entail one or more agents (human or computational) that perform a partially ordered set of tasks using tools that consume resources to produce intermediate or final products.

control the use of the concepts and tools for modeling, enacting, discovering, and recovering knowledge work processes. The *mechanisms and tools* will embody and support these concepts and techniques.

In addition, these results lay the foundation for a comprehensive, integrated knowledge work environment that integrates process modeling, enactment, discovery, and recovery with emerging capabilities for process/resource simulation, visualization, scheduling and allocation/deployment being investigated elsewhere. The design and demonstration of such an environment is an appropriate candidate for a follow-on research investigation. Beyond this, many of our prior research results have been transferred into products that have been commercialized by a variety of firms. Thus, we expect to transition our results to future research investigations and eventually to commercial applications.

Research Approach

We have been involved in systematically observing, modeling, scheduling, integrating, and enacting complex organizational processes for more than 10 years [cf. MS90, SM97, NS91, NS99, NS01, S98, S02b, SN97]. Most of this prior effort has focused on examining and engineering the processes involved in large-scale software system development for commercial, military, or academic applications. For example, we have recently focused on software development processes within globally dispersed virtual enterprises [NS99], particularly those developing open source software [S02a] with centralized corporate sponsorship or control[2]. These processes can be characterized as entailing:

- The production and consumption of knowledge based products (e.g., software programs, development artifacts, documentation, project management reports, and MIME object types)
- The acquisition of information (end-user requirements, software test case evaluation, bug/defect reports from end-users, etc.) from which knowledge products are built
- Having customers with a high level of urgency for these knowledge products
- Reliance on worldwide information sources and repositories that may not be owned or readily controlled

---

[2] Examples of large projects of this kind include the OpenOffice.org and NetBeans projects sponsored by SUN Microsystems Inc., and the Mozilla.org project sponsored by Netscape and AOL. Models of selected processes for these projects have already been captured and coded using traditional methods [CLC02, ONHJ02]. Another dozen or so small/mid-size open source software projects with centralized corporate sponsors are identified elsewhere [S02c]. Beyond these, organizations like *infoDev*, the Information for Development Program of the World Bank, appear interested in encouraging research, development, and policy studies of open source software for centrally controlled electronic government applications. Finally, large international financial institutions like Barclays Global Investors (BGI) and Dresdner Kleinwort and Wasserstein (DKW) are also centrally controlled enterprises that have invested in and rely on open source software development processes and collaborative development environments to support their global enterprise information systems development projects and operations [S02c].

- A high level of dependence on personnel (software developers) with specialized (application domain specific) expertise and expensive skills
- Access to a worldwide IT infrastructure (Internet, Web, SourceForge.net, etc.) to support product development.

We have also focused attention on the *life cycle engineering of complex organizational processes* [S98, SM97, SN97] for corporate financial operations, telecommunications systems design, military procurement and system acquisition, research grants management, feature film production, interactive teleradiology, and others. Thus we bring an extensive history of prior research results and experience in approaching the problem managing distributed enterprise processes associated with knowledge intensive dynamic systems.

Our approach to the overall problem of how to manage knowledge intensive dynamic systems is process centered. Problems of modeling, analyzing, simulating, integrating, and enacting routine enterprise processes with dynamic or hidden workflows, and how they may be associated with (semantic) hypertext webs of knowledge products, information assets and other repositories, are well known to us [NS91, NS99, NS01, S00, SN97] and others [e.g., HW99, LRS02, SHC01].

One thing we have learned is that *explicit models of complex processes* can directly contribute to continuous process improvement, process redesign, mitigate common process breakdowns, automate Web-based process enactment, and more [cf. NS99, NS01, S98, S00, S01b, S02b, SN97]. These capabilities in turn can lead to dramatic improvements (4X-20X reductions in process cycle time) in process efficiency and effectiveness, as well as cost savings [S01a]. However, acquiring the requisite organizational and process domain knowledge needed to create an explicit high quality process model is a slow, labor intensive endeavor. As a result, we have come to find that it is often nearly as effective to develop and engineer *low-fidelity models*[3] of routine and dynamic knowledge work processes. These simpler models serve as the "seed" from which adaptive process descriptions, proscriptions, or prescriptions can be modeled, grown, repaired, redesigned, and continuously improved.

Our focus in the proposed research effort is to investigate how to configure and rapidly reconfigure process control structures and resources when employing low-fidelity process models. This leads to three lines of study.

- We need to develop scaleable techniques for *modeling, enacting, and capturing* globally dispersed complex enterprise processes that use low-fidelity process models to coordinate and deploy (allocate) access to distributed resources by process users. Our focus is to develop mechanisms and system architectures that can enact low-fidelity process models, together with a capability to capture process and resource event histories in a form suitable

---

[3] Process models and process enactment instances are similar to abstract plans and instantiated plans found in knowledge-based systems [cf. MS99, SM96]. Low-fidelity process models are thus similar to abstract (or reusable) plans. Process models much like plans, are typically associated with resource allocation and scheduling system capabilities that support complex enterprise activities [MS93, MS99, SHC01, SM97].

for use in process discovery and recovery tasks.

- We need to develop scaleable techniques for *discovering* process control structures within emergent or hidden workflows that can serve as low fidelity process models. These models are created by transparently capturing and generalizing the history of events, conditions, and contexts associated with process enactment activities that create, update, delete, or associate (i.e., hyperlink) online object types or knowledge-based products.

- We need to develop scaleable techniques for *recovering and articulating* the process control structures that breakdown or fail during a routine or dynamic process enactment. Prior research has shown that weakly structured or constraint-relaxed process control structures are prone to breakdown or fail during enactment [BS89, MS93, SM97].

Each of these three investigations is envisioned to leverage and expand the process modeling and enactment framework that we have been investigating for the past few years [NS90, NS99, NS01, SN97]. This entails a collaborative research project (via subcontract) lead by Walt Scacchi (PI, UCI) and John Noll (Co-PI, SCU). Scacchi is leading the effort on process discovery and recovery techniques, while Noll is leading the effort on new techniques and mechanisms for modeling, coordinating, enacting and monitoring (for capture of enactment event histories of) complex enterprise processes.

The following sections briefly elaborate each of the three lines of study identified above.

Modeling, enacting, and capture of globally dispersed enterprise processes

Prior research has investigated how best to specify enterprise processes in sufficient detail to provide some form of active or Web-based process management support [e.g., HW99, LRS02]. In general, Web-based approaches focus effort on encoding work processes as programs that access and manipulate Web-based resources and services. Unfortunately, the power and adaptation of these programs is outside the realm of experience, skill set, or emergent needs of the users who must work with such systems. Instead, we envision a globally dispersed knowledge work environment where dynamic enterprise processes are more transparent, easy to modify, and adaptable by users, whether with or without the support of process programmers. This requires processes that can be both described and interpreted as high-level models [NS01], rather than lower-level workflow programs, middleware, or Web services [LRS02].

This leads us to propose a system to facilitate communication and collaboration among knowledge workers to disseminate process expertise as widely as possible. In this approach, users in different process roles are given high-level guidance (or generalized plans) about what activities to perform or what objectives to achieve, and how to perform them. Users should thus be free to carry out the details of those activities through process enactments that are consistent with or adapt to their expertise. Achieving this entails development of a globally dispersed, model-driven process enactment environment that integrates:

- A *distributed process deployment and execution mechanism* for enacting low fidelity process models. Our prior research experience [BEF02, NB02, NS99, NS01, SN97] suggests that these models can be easily specified or generated. Since low fidelity process models specify the minimal aspects of a work process (e.g., required and/or provided resources, appropriate tools, user roles, and proscribed activities), these models also tend to be stable, reusable, and reconfigurable, as well as enactable via navigational browsing [NB02, NS99, NS01].

- A *virtual repository of artifacts* [cf. NS91, NS99] providing access to distributed collections, repositories, and databases of information objects related to the work to be performed. Information resources (documents, images, diagrams, databases, etc.) will be found in globally dispersed repositories of different types with locally autonomous access and update procedures, that users will want to browse, manipulate, or hyperlink [BEF02, NS91, HW99, NS99, NS01].
- A *data capture facility* for monitoring, collecting, recording, and replaying resource and process enactment event histories [cf. CW98, SM97], that supports process discovery and recovery analyses. Continuous process improvement or emergent process redesign requires knowledge about who did what in a process enactment and shared resource space, when, where, how and why. Getting users to provide this information is too much effort, but providing an automated mechanism to capture (and replay) the data/events would alleviate such effort.

Overall, the model-driven approach to process modeling, enactment, and event capture provides a foundation for process discovery and recovery investigations. In addition, this approach establishes a foundation and dispersed process work environment that can subsequently integrate simulation, visualization, resource allocation/deployment, and scheduling mechanisms into a comprehensive Web-based environment in a follow-on study. As such, our approach to model-driven process enactment and event capture is a significant departure from existing approaches to workflow automation or Web-based services for process automation [cf. LRS02].

Discovering processes from routine, dynamic, or hidden workflow instances

As already noted, acquiring the requisite organizational and process domain knowledge to construct high fidelity process models is valuable, but costly and time-consuming. We need to be able to rapidly construct or (re)configure process models in a less costly, less time consuming, and more transparent manner. In a complex enterprise setting where routine, dynamic or hidden knowledge work processes occur, process enactments will be emergent and reactive, rather than procedural and planned in detail. Furthermore, when process enactments are physically distributed but logically centralized for scheduling and control purposes [NS99], then we need a more innovative technique than traditional for process modeling, analysis or simulation [cf. MS90, S98, SM97].

Prior research in this area has applied grammatical inference, Markov modeling, or temporal constraint ordering techniques to identify process fragments from partially ordered or time-stamped event records to identify or generate process models [CW98, HY03]. Use of Web-based resource usage or process enactment [HW99, LRS02, NS99] event streams has not been explored for discovery purposes, nor have other techniques from knowledge discovery approaches [FPS92]. However, our approach is to employ, evaluate, and refine these kinds of capabilities. For example, in our work, we have experimented with the creation of Web-based process modeling and enactment mechanisms [NS99, NS01, SM97, SN97] that capture process enactment events associated with the manipulation of globally distributed information resources and computing services. Figure 1 displays a screenshot of a step in a procurement process that reveals contextual information and process enactment event history, which is not found in command shell histories [NS01].

We will investigate, evaluate, and develop new techniques for discovering patterns of local-to-global resource and process enactment events, actions, and conditions that manipulate information resources or artifacts (e.g., creating, updating, deleting, or hyperlinking information objects, diagrams, messages, files, Web sites, or reports). This will also require automatically capturing aspects of the process enactment context (e.g., human roles, tools invoked, repositories accessed, network host addresses, and event timestamps) that are attributes of the activities or artifacts involved. We anticipate that we will be able to automatically find process control fragments that are interspersed among a longer set of situated but coincidental (i.e., not process related) actions. Process resource usage events and process execution event histories will be evaluated as one direction for determining what process information can be gathered automatically and transparently.

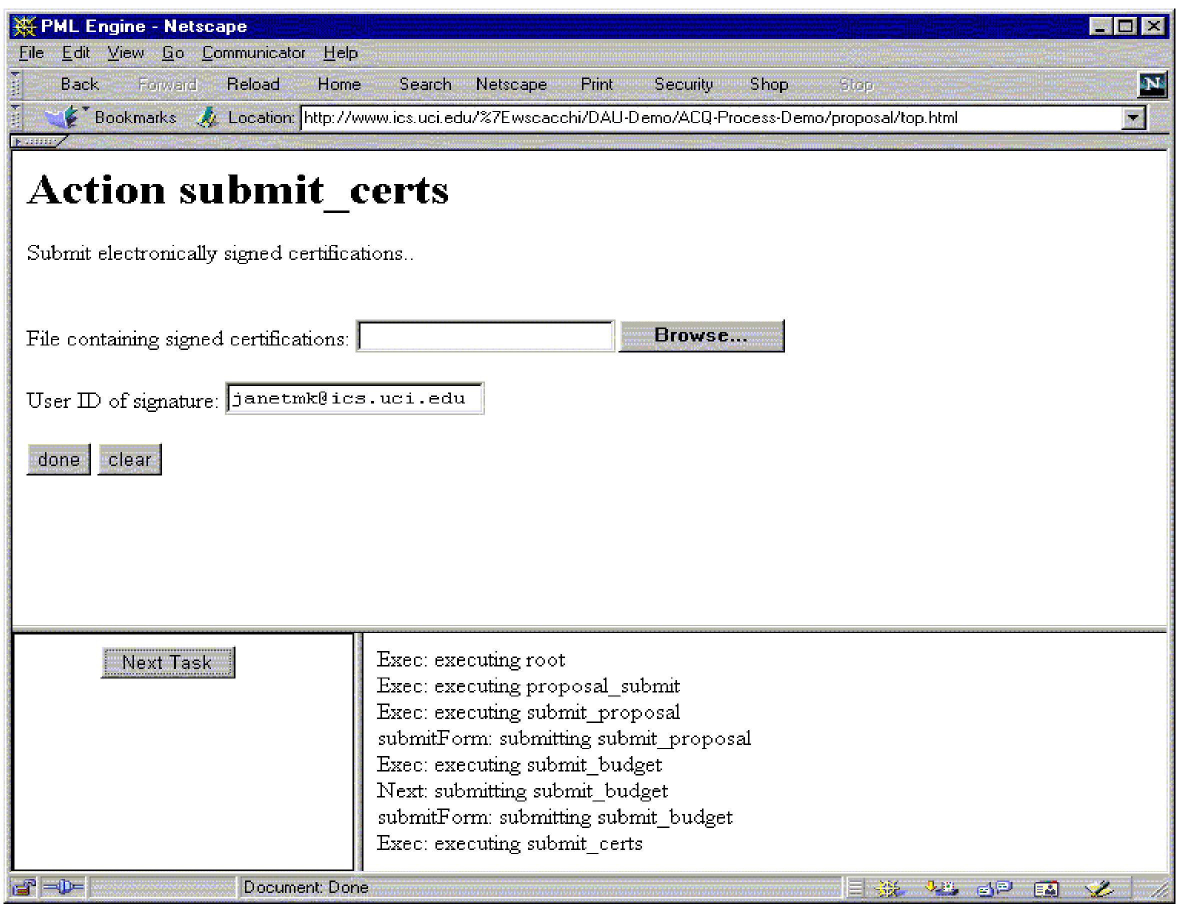

Figure 1. Enactment of a Web-based procurement process with captured history (lower right frame) [Noll and Scacchi 2001, Scacchi 2001a].

Recovering and articulating processes that breakdown or fail during their enactment

Process control structures both shape, and are shaped by, teamwork structures [BS89]. Low-fidelity process models can specify what a process needs to accomplish, but the exigencies of resources, knowledge products, and team membership at hand, give rise to unanticipated conditions or events that must be resolved for work to proceed. These exigencies and unanticipated conditions help characterize how the process enactment at hand is breaking down or failing [MS93, SM97]. These process breakdowns or failures are a primary, recurring cause of business process dynamics [SM97].

In related research, we have developed an approach for diagnosing, replanning, and rescheduling process enactments that breakdown or fail during enactment. This approach is called "articulation" [MS93], and Figure 2 below provides an overview of process enactment repair and recovery control scheme that employs articulation.

Our prior effort with articulation was focused on reasoning about how centralized software engineering processes could be repaired and recovered once they failed. In the proposed effort, we will focus attention on physically distributed but logically centralized processes for knowledge work that utilize globally distributed resources and IT infrastructure. This approach stands in contrast to related efforts that seek to provide Web-based process enactment support [HW99, LRS02] under the assumption that those processes will neither fail in practice, nor need to be adapted, reconfigured, or redesigned on demand. Nonetheless, our approach to articulation appears to be more closely aligned to those efforts that embrace more of a mixed initiative of system provided guidance and user-driven adaptation (e.g., for process enactment planning, resource allocation and task scheduling) that supports a globally dispersed community of knowledge workers through a Web-based environment [cf., BEF02, NS99, MS93, SCH01, SM97].

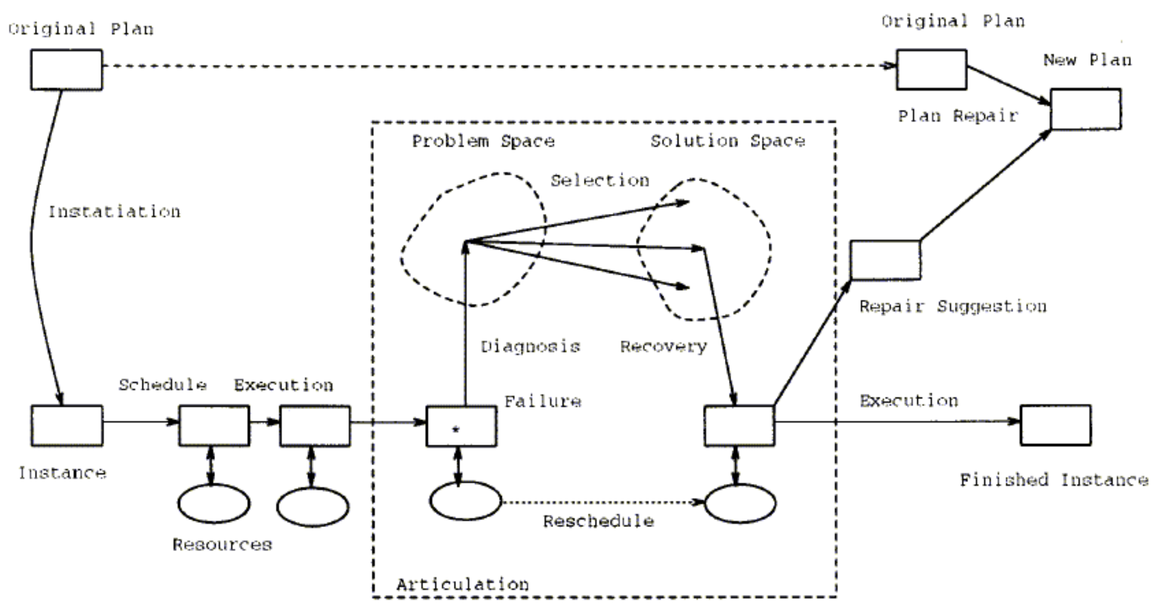

Figure 2. An overview of a knowledge-based approach to diagnosing, replanning and rescheduling process plans that breakdown or fail during enactment [Mi and Scacchi 1993].

References

[BS89] S. Bendifallah and W. Scacchi, Work Structures and Shifts: An Empirical Analysis of Software Specification Teamwork, *Proc. 11th. Intern. Conf. Software Engineering*, IEEE Computer Society Press, Pittsburgh, PA. 260-270, (May 1989).

[BEF02] M. Beiber, D. Engelbart, R. Furuta, S.R. Hiltz, J. Noll, J. Preece, E. Stohr, M. Turoff, and B. Van De Walle, Towards virtual community knowledge evolution, *J. Management Information Systems*, 18(4), 11-36, 2002.

[CLC02] B. Carder, B. Le, and Z. Chen, The Process of Quality Assurance and in the Mozilla Project Release Cycle, Internal project report, Institute for Software Research, June 2002.

[CW98] J.E. Cook and A.L. Wolf, Discovering Models of Software Processes from Event-Based Data, *ACM Trans. Software Engineering and Methodology*, 7(3), 215-249, July 1998,

[FPM92] W.J. Frawley, G. Piatetsky-Shapiro, C.J. Matheus, Knowledge Discovery in Databases: an Overview, *AI Magazine*, 13, 57-70, 1992.

[HW99] J.M. Haake and W. Wang, Flexible support for business processes: extending cooperative hypermedia with process support, *Information and Software Technology*, 41, 355-366, 1999.

[HY03] S.-Y. Hwang and W.-S. Yang, On the discovery of process models from their instances, *Decision Support Systems*, 34(1), 41-57, 2003.

[LRS02] F. Leymann, D. Roller, and M.-T. Schmidt, Web services and business process management, *IBM Systems J.*, 41(2), 198-211, 2002.

[MS90] P. Mi and W. Scacchi, A Knowledge-Based Environment for Modeling and Simulating Software Engineering Processes, *IEEE Trans. Data and Knowledge Engineering*, 2(3), 283-294, September 1990. Reprinted in *Nikkei Artificial Intelligence*, 20(1), 176-191, January 1991, (in Japanese). Reprinted in *Process-Centered Software Engineering Environments*, P.K. Garg and M. Jazayeri (eds.), IEEE Computer Society, 119-130, 1996.

[MS93] P. Mi and W. Scacchi, Articulation: An Integrated Approach to the Diagnosis, Replanning, and Rescheduling of Software Process Failures, *Proc. 8th. Knowledge-Based Software Engineering Conference*, Chicago, IL, IEEE Computer Society, 77-85, 1993.

[MS96] P. Mi and W. Scacchi, A Meta-Model for Formulating Knowledge-Based Models of Software Development, *Decision Support Systems*, 17(4), 313-330, 1996.

[MS99] K. Myers and S. Smith, Issues in the Integration of Planning and Scheduling for Enterprise Control, *Proc.DARPA Symposium on Advances in Enterprise Control,* 1999.

[NB02] J. Noll and B. Billinger, Modeling coordination as resource flow: An object-based approach, *Proc. 2002 IASTED Conf. Software Engineering Applications*, Cambridge, MA, November 2002.

[NS91] J. Noll and W. Scacchi, Integrating Diverse Information Repositories: A Distributed Hypertext Approach, *Computer*, 24(12), 38-45, December 1991.

[NS99] J. Noll and W. Scacchi, Supporting Software Development in Virtual Enterprises, *Journal of Digital Information*, 1(4), February 1999.

[NS01] J. Noll and W. Scacchi, Specifying Process-Oriented Hypertext for Organizational Computing, *J. Network and Computer Applications,* 24(1), 39-61, 2001.

[ONHJ02] M. Oza, E. Nistor, S. Hu, and C. Jensen, NetBeans Requirements and Release Processes, Internal project report, Institute for Software Research, June 2002.

[S98] W. Scacchi, Modeling, Integrating, and Enacting Complex Organizational Processes, appears in K. Carley, L. Gasser, and M. Prietula (eds.), *Simulating Organizations: Computational Models of Institutions and Groups*, 153-168, MIT Press, 1998.

[S00] W. Scacchi, Understanding Software Process Redesign using Modeling, Analysis and Simulation, *Software Process--Improvement and Practice*, 5(2/3), 183-195, 2000.

[S01a] W. Scacchi, Redesigning Contracted Service Procurement for Internet-based Electronic Commerce: A Case Study, *J. Information Technology and Management*, 2(3), 313-334, 2001a.

[S01b] W. Scacchi, Modeling and Simulating Software Acquisition Process Architectures, *J. Systems and Software*, 59(3), 343-354, 15 December 2001b.

[S02a] W. Scacchi, Understanding the Requirements for Developing Open Source Software

Systems, *IEE Proceedings--Software*, 149(1), 2002a.

[S02b] W. Scacchi, Process Models in Software Engineering, in J. Marciniak (ed.), *Encyclopedia of Software Engineering* (Second Edition), 993-1005, Wiley, New York, 2002b.

[S02c] W. Scacchi, Open EC/B: A Case Study in Electronic Commerce and Open Source Software Development, Final Report, NSF CRITO University-Industry Consortium, July 2002c.

[SM97] W. Scacchi and P. Mi, Process Life Cycle Engineering: A Knowledge-Based Approach and Environment, *Intern. J. Intelligent Systems in Accounting, Finance, and Management*, 6(1), 83-107, 1997.

[SN97] W. Scacchi and J. Noll, Process-Driven Intranets: Life Cycle Support for Process Reengineering, *IEEE Internet Computing*, 1(5), 42-49, 1997.

[SHC01] S. Smith, D. Hildum, and D.R. Crimm, Toward the Design of Web-Based Planning and Scheduling Services, *Proc. ECP-01 / Planet Workshop on Automated Planning and Scheduling Technologies in New Methods for Electronic, Mobile and Collaborative Work*, September, 2001.

[VS99] A. Valente and W. Scacchi, Developing a Knowledge Web for Business Process Redesign, Presented at the *1999 Knowledge Acquisition Workshop*, Banff, Canada, October 1999.