UPGRADE
The European Journal for the Informatics Professional
http://www.upgrade-cepis.org
Vol. V, No. 5, October 2004

UPGRADE is the European Journal for the Informatics Professional, published bimonthly at <http://www.upgrade-cepis.org/>

UPGRADE is the anchor point for **UP**ENET (**UP**GRADE European NETwork), the network of CEPIS member societies' publications, that currently includes the following ones:
- Mondo Digitale, digital journal from the Italian CEPIS society AICA
- Novática, journal from the Spanish CEPIS society ATI
- Pliroforiki, journal from the Cyprus CEPIS society CCS
- Pro Dialog, journal from the Polish CEPIS society PTI-PIPS

Publisher
UPGRADE is published on behalf of CEPIS (Council of European Professional Informatics Societies, <http://www.cepis.org/>) by Novática <http://www.ati.es/novatica/>, journal of the Spanish CEPIS society ATI (Asociación de Técnicos de Informática <http://www.ati.es/>).
UPGRADE is also published in Spanish (full issue printed, some articles online) by Novática, and in Italian (abstracts and some articles online) by the Italian CEPIS society ALSI <http://www.alsi.it> and the Italian IT portal Tecnoteca <http://www.tecnoteca.it/>.
UPGRADE was created in October 2000 by CEPIS and was first published by Novática and INFORMATIK/INFORMATIQUE, bimonthly journal of SVI/FSI (Swiss Federation of Professional Informatics Societies, <http://www.svifsi.ch/>).

ISSN 1684-5285

Next issue (December 2004):
"Cryptography"
(The full schedule of **UP**GRADE is available at our website)

* This monograph will be also published in Spanish (full issue printed; summary, abstracts and some articles online) by **Novática**, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*) at <http://www.ati.es/novatica/>, and in Italian (online edition only, containing summary abstracts and some articles) by the Italian CEPIS society ALSI (*Associazione nazionale Laureati in Scienze dell'informazione e Informatica*) and the Italian IT portal Tecnoteca at <http://www.tecnoteca.it>.

Managing Distributed Projects in GENESIS

Lerina Aversano, Andrea De Lucia, Matteo Gaeta, Pierluigi Ritrovato, and Maria-Luisa Villani

The success of large software projects conducted by different organization sites may be determined by the inter-site coordination and cooperation of the working teams, thus automated support to distributed project management can be useful. In this context we present the GENESIS (Generalized ENvironment for procESs management in cooperatIve Software engineering) approach to distributed process modelling and enactment, realized through an event dispatching architecture whose distinctive feature is a decentralized and autonomous definition of the multi-site software processes.

Keywords: Coordination and Cooperation, Distributed Process, Large Software Projects, Multi-Site Software Projects, Project Management, Process Modelling, Software Processes, Workflow Management.

1 Introduction

Software projects are generally complex and may be distributed across sites, so they require the coordination and cooperation of teams of software engineers from different geographical locations and possibly belonging to different organizations. This scenario is now becoming commonplace as a result of globalization and therefore automated tools for the management and execution of these projects are highly desirable [15][4][10]. In fact, the support for distributed project management is a relevant problem for two reasons.

- projects may involve a big number of concurrent activities, which impose an adequate coordination support to keep them on track and under control;
- when a project spans multiple sites that generally work in a largely autonomous manner, these sites will not necessarily be following the same process models, nor they will be all employing the same methods and tools. Hence, it may be required that, whenever it is possible, parts of the project should be under the responsibility of local project managers,

Lerina Aversano received a BSc in Computer Engineering from the *Università degli Studi del Sannio*, Benevento, Italy, in 2000, and a PhD in Computer Engineering from the same university, in 2003. She is currently an assistant researcher at the RCOST (Research Centre on Software Technology) of the mentioned university. Her research interests include process and information system modelling, workflow management, document management, business process reengineering, software reengineering and migration. <aversano@unisannio.it>

Andrea De Lucia received a PhD in Electronic Engineering and Computer Science from the *Università degli Studi di Napoli*, Naples, Italy, in 1996. He is an Associate Professor of Software Engineering at the Department of Mathematics and Informatics of the *Università degli Studi di Salerno*, Italy. His research interests include software maintenance, reverse engineering, visual languages, workflow management, and document management. He is the program co-chair of the 11th IEEE Working Conference on Reverse Engineering (WCRE 2004) and of the 3rd Workshop on Cooperative Supports for Distributed Software Engineering Processes (CSSE2004). Prof. De Lucia is a member of the IEEE and the IEEE Computer Society. <a.delucia@unisa.it>

Matteo Gaeta is Researcher for Systems of Information Processing and University Teacher of Information Fundaments at the Faculty of Engineering of the *Università degli Studi di Salerno*, Italy. He is Director and Member of Board of Directors of CRMPA (Centre of Research in Pure and Applied Mathematics), a no-profit Consortium between industrial enterprises and the university; he is also Deputy Director of the Centre of Excellence on Learning and knowledge of the same university. He is a member of the Technical Committee by the Italian Minister of Productive Activities. He is a member of the Steering Committee of the Italian Ministry of Agricultural and Forest Policies for the consultancy and the scientific, organisational support to research, development and innovation. <gaeta@crmpa.unisa.it>

Pierluigi Ritrovato is Research and Technology Director of CRMPA (Centre of Research in Pure and Applied Mathematics), *Università degli Studi di Salerno*, Italy. He has participated in, or coordinated, several Research and Development projects either Italian or European on themes relative to: distributed systems and architecture, <http://www.elegi.org>; GRID technologies and architectures, <http://www.eu-grasp.net>; Distributed Software Engineering, <hhtp://www.genesis-ist.org>; application interoperability and cooperation; CSCW, and Internet Technologies. <ritrovatog@crmpa.unisa.it>

Maria-Luisa Villani is a Research Assistant at RCOST (Research Centre on Software Technology) of *Università degli Studi del Sannio*, Benevento, Italy. She graduated (cum laude) in Mathematics at the University of Napoli in 1994. From September 1995 to September 1999 she was at the University of Warwick – England, where she obtained a MSc degree (1996) and a PhD (1999) in Matematics. In September 2001 she got a Master degree in Software Technology at the *Università degli Studi del Sannio* and joined RCOST. Her current research interests are service-centric software engineering, software process technology and modelling, model checking techniques for program verification. <villani@unisannio.it>

who can better organize the local activities (i.e., processes and sub-processes) and resources involved.

In particular, the latter issue poses problems concerned with decentralized and autonomous modelling of multi-site software processes. Most works on distributed process management focus on developing paradigms and architectures for the enactment of these processes but they scarcely address decentralized modelling [16][14][13][9][4][10][7][8]. In most cases process modelling is a centralized activity and the enactment of portions of the process is distributed among different workflow engines. In some cases, the central process model is collaboratively edited with the contribution of people from different sites [10] and each site has visibility of the overall process model. A different approach is used in OzWeb [4] where process models are autonomously defined on the different sites and cooperate through specifically designed interfaces.

In this paper we present the GENESIS (Generalized ENvironment for procESs management in cooperatIve Software engineering) approach to distributed process modelling. The GENESIS platform is an outcome of a research project aimed at designing and developing a non-invasive and open source system to support software engineering processes in a highly distributed environment. In GENESIS, the global view of the project is modelled and enacted at the coordinator site (that is the technical leader of the distributed software project [14]), while sub-processes can be autonomously modelled and executed on different organizational sites. The global process model can be collaboratively edited by the project managers of the different sites.

The paper is organized as follows. Section 2 discusses related work. Section 3 presents an overview of GENESIS, with particular reference to the flexible approach adopted for distributed process modelling and enactment. Section 4 details an asynchronous communication protocol for distributed project definition and management, while Section 5 concludes.

2 Related Work

In global and virtual enterprises, software processes consist of multiple sub-processes that may span over organizational boundaries. The current commercial workflow technology does not provide the necessary functionality to model, enact, and manage distributed processes due to its mostly centralized server architecture. Numerous are coordination functionalities that cannot be fulfilled by traditional workflow systems [14][5], such as the support for distributed execution of a workflow; shared access to data and the use of groupware tools.

Modern workflow management systems exploit the web as a mean to enable distributed access to the facilities provided by the workflow engine [1][12][5][15]. However, most of these systems are still based on a client-server architecture and the problem of designing architectures for distributed process modelling and enactment of the process is still a research issue [16][14][13][9][4][10][7][8][17]. PROSYT is an artifact based PSEE [7]. Each artifact produced during the process is an instance of some artifact type, which describes its internal structure and behaviour. All the routing in this model is based on the artifact and the operations on them. PROSYT also allows for distributed enactment facilitated by an event-based middleware [8].

In [9] the authors propose an approach for the distributed execution that exploits an event notification service, named READY. Workflow participants, both workflow engines and agents, can subscribe to events that trigger the start of workflow activities and processes, and events that describe state changes in the workflow processes they are interested in. Therefore, the configuration of the participants in a workflow can be dynamically changed without requiring any modifications to the existing architecture.

The Endevors project [13] proposes an approach to provide a coordination mechanism for distributed process execution and tool integration by using the Hypertext Transfer Protocol (HTTP). The system uses a layered object model to provide for the object-oriented definition and specification of process artifacts, activities, and resources. The intent for distribution is to support a wide range of configurations with varying degrees and kinds of distribution: stand-alone with a base system configuration without distributed components, multi-user with a single remote data-store, multi-user with a single remote data-store are the configuration experimented for distribution.

Kötting and Maurer [14] propose an extension of MILOS [14] which focuses on the process support for virtual corporations. They propose three different approaches for distributed process enactment: replicated workflow engines; central coordinator site, and a peer-to-peer architecture for data exchange. The authors do not address the problem of decentralized process modelling.

Grundy et al. [10] focus on distribution problems in process modelling. The proposed system provides mechanisms for collaboratively editing process models both in a synchronous and an asynchronous way, together with version management support. The architecture is based on a central site maintaining the process model and distributed sites enacting portions of that model.

In the Ozweb environment the peer-to-peer paradigm for distribution is adopted [4]. Here a decentralized system consists of independent sub systems spread among multiple sites. In particular, the authors focus on the process autonomy of each sub system that should be self contained and operationally independent. To this aim they introduce the concept of 'treats' to guarantee compliance of the artifacts exchanged between sub-processes.

Our approach mixes both these features: we have the notion of a coordinator site where a global process can be defined in a collaborative way by the project managers of the different cooperating sites of a virtual organization. Sub-processes executed on different sites are autonomously defined and only have to respect the interfaces defined at global level. Moreover, the depth of the global process model is not limited to just two levels, as it is possible for a partner of the virtual corporation to have further sub-contractors.

3 Distributed Process Management in GENESIS

Traditional workflow management systems do not provide adequate support for the evolution of software organi-

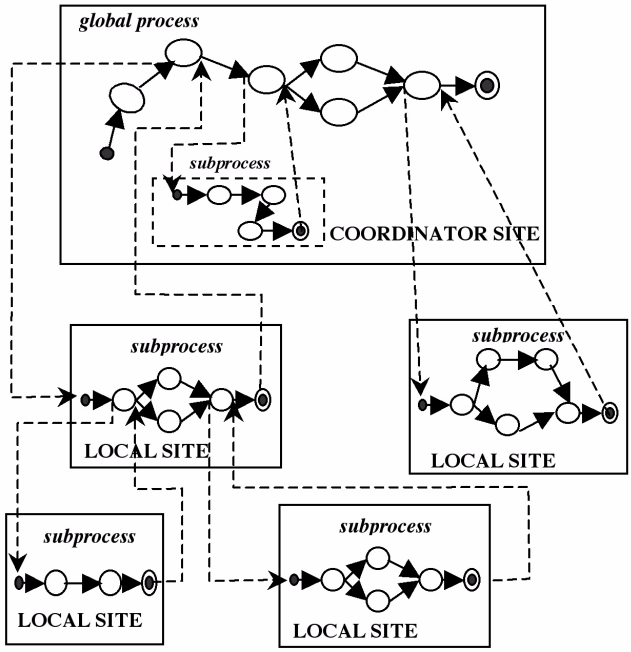

Figure 1: Hierarchical Process Decomposition.

zations towards distributed virtual models. The main open problem remains the systematic definition of distributed process models and their enactment across multiple sites using appropriate abstractions and mechanisms. The GENESIS environment has been developed with the aim to provide solutions to these problems; to this end the environment provides a special support for a distributed scenario, from the modelling of a distributed process to its enactment.

Distributed projects in GENESIS require a *project coordinator site*, managing the overall project and a number of local sites, managing specific project workpackages. The coordinator is in charge of modelling and executing the global process for the project, while the local sites are in charge of defining and executing sub-processes of that model, and concerning their own workpackages (see Figure 1).

The architecture of each GENESIS site includes different components as depicted in Figure 2:

- a workflow management system to model and enact software processes;
- an artifact management system to store and retrieve the artifacts produced within a process;
- a resource management system to allocate resources, in particular human resources, to a project;
- an event engine and communication system to collect and dispatch events raised during process management, such as the termination of an activity or the production of an artifact;
- a metric engine in charge of collecting metrics and presenting synthetic reports about the project status.

3.1 Process Modelling in GENESIS

In order for companies to use, to a certain extent, their existing practices and to ensure the quality of the overall process, support is given both for the top-down and bottom-up definition of processes. These may be achieved through a *multi-level process definition*, where an activity at one level (i.e. a super-activity) may correspond to a sub-process at lower level: the activity may be assigned to a different site, where it is independently modelled in a decentralized manner, and executed. In fact, the only requirement is that the sub-process interface (in terms of input / output artifacts) is conform to that of the super-activity (see Figure 1). This process componentization also enables integration of (sub)process models in a bottom-up fashion.

Besides super-activities, a process model can also contain global activities, i.e. activities that can be collaboratively performed by workers from different sites. The project manager of each site who participate in a global activity is in charge of providing the needed human resources for the activity.

In GENESIS the process modelling language is the same both at global level, to model the global software process with the coordination of the composing sub-processes, and at local level, to model the sub-processes at the single GENESIS sites. In order for the system to be the least invasive, a coarse grained definition of the process elements (activities and artifacts) has been decided. Activities are essentially described by the artifacts they will produce, and freedom is left to the worker(s) to decide how to actually perform them, in accordance with the organization standards.

GENESIS provides two process modelling stages. A process designer may create abstract process models, through the process definition tool, according to standards of the specific organization. Abstract process models include description of activities (including roles of people performing an activity and types of input and output artifacts) and enactment rules (or transitions) that basically describe control and data (artifact) flow between activities and are expressed through the Event-Condition-Action (ECA) paradigm. Abstract process models have to be customized for each project by specifying project data, such as the actual people performing activities and the actual artifacts. Project managers can use the abstract process definitions as templates to create concrete process models, through the project management tool as discussed in Section 4. These models can then be executed by the workflow enactment service, and different process instances may be enacted from the same concrete process model. Further details concerning the process modelling language are described in [3].

3.2 Distributed Process Enactment in GENESIS

The need to avoid a big bang approach to process modelling has been a guideline for the design of the GENESIS environment. As underlined by the industrial partners of the project, the usability of the environment could be significantly impacted by a rigid approach to process modelling and enactment. Indeed, an incremental approach at both activity and process definition level is required for the management of real projects. Often the start up of a project precedes a complete definition of the process models to be used for supporting and controlling it. In fact, for some scenarios, it may happen that the big picture of the process cannot be completed at design time because

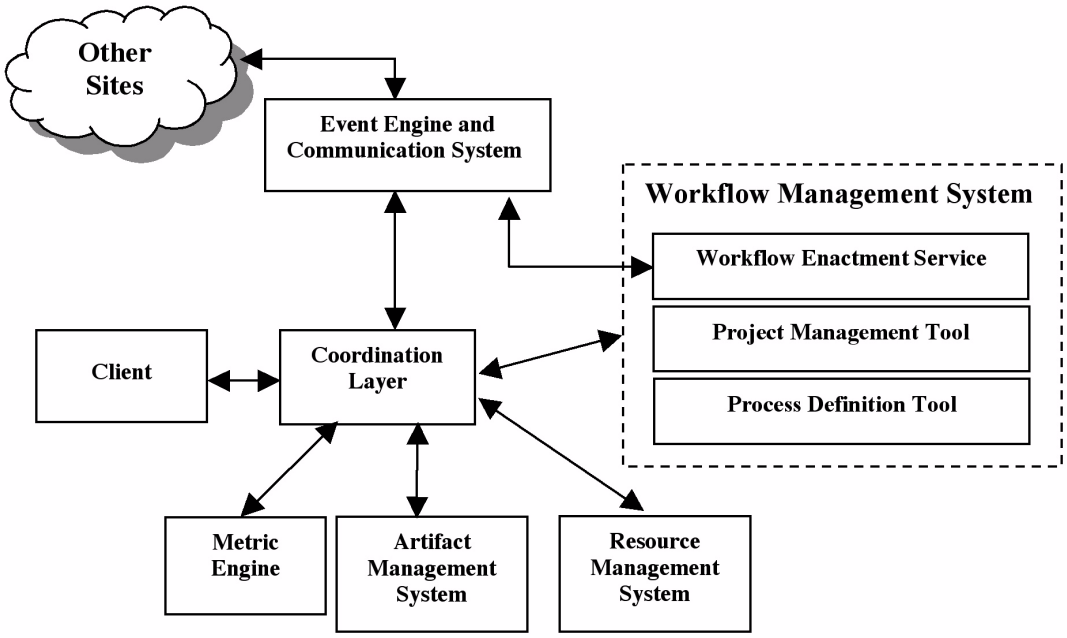

Figure 2: GENESIS Site Architecture.

there is not enough information for that. Thus, it is realistic to assume that the process models to be followed should be incrementally defined: starting from a simple model and 'complicating' it on the basis of the arising needs. In this respect, the GENESIS workflow engine is able to enact rough defined processes and facilities are offered to the project manager by the project management tool to refine the process definition at runtime.

Indeed, the model concretization process may be incremental as the process can start as soon as the needed resources have been assigned to the initial activity, and independently of the rest. In fact, checks are made during enactment to make sure that each activity has its resources allocated when it needs to be started. The project manager will be notified to staff the activity in order for the process to proceed. Most importantly, process may be dynamically modified. In fact, facilities are provided by the platform to both:

- change properties of the process instance, without changing the process model definition (i.e. the process map does not change). For example, a change of an activity assignment property;
- restructure the process map at run-time (e.g. adding/deleting activities), in order for the process instance to better match the real process.

These facilities are especially useful to handle unforeseen exceptions. In this respect, it should be noted that the distribution and decentralization of the process model through super-activities allows to restrict the scope of exceptions within the local sites, as long as they do not impact the border with the upper level process.

4 An Asynchronous Protocol for Distributed Project Definition

In GENESIS an asynchronous protocol has been defined for the communication between the global coordinator level and the local coordinated sites during the instantiation of a distributed software project. We distinguish three main phases: the creation of the project both at the coordinator site and the local sites, where the resource managers associate people to the project and select the project managers; the definition of the global process involving project managers of the different sites; and the definition of the local processes, independently defined by the different local project managers.

4.1 Project Creation

At the start of a new project the resource manager of the global site creates a project using the resource management tool of the platform. This means that s/he selects the human resources allocated on the global site and the local sites participating to the project. The allocated resources of the global site and in particular the global project manager are notified through the event engine and communication system.

A "Global Project Creation" event is also sent to the involved sites. Each resource management system of a local site automatically stores the received event and notifies the local resource manager. At this point, the local resource manager decides the allocations of the human resources and the local project manager who are notified about. The event "Local Project Creation" is sent to the resource management tool of the coordinator site, to store the needed information at global level and notify the global project managers.

4.2 Global Process Definition

Once the global project manager has received the notification concerning the "Global Project Creation", s/he can start defining the needed concrete process models for the project, starting from available abstract process models (if a suitable abstract process model is not available, it has to be created first). The global process model includes super-activities to be assigned to local sites and global activities, carried out by groups of people distributed among different sites. Local project managers can collaborate with the global project manager for the definition of the global process, as soon as they are selected by the local resource managers.

Each super-activity has to be assigned by the project manager to a site participating in the project. In this case a "Super Activity Creation" event is sent to the local site together with information concerning the super-activity (start and end date, artifact types, etc.). The project management tool of the local site automatically stores and associates this information to the corresponding project. The event is also notified to the project manager of the local site, as soon as s/he is appointed.

For each global activity, the global project manager selects the sites that have to provide human resources to collaboratively work on the activity (examples of such activities are project reviews to be conducted by the global and local project managers). A "Global Activity Creation" event is sent to each site involved in the global activity together with the role and number of required people.

It is worth noting that a concrete global process can start independently of the local process definition status (see next sub-section).

4.3 Local Process Definition

For each super-activity assigned to a local site, the project manager creates the corresponding concrete local process model (again, starting from an available abstract process model). As soon as the enactment of the concrete local process can start, a "Local Process Model Creation" event is sent to the project management tool of the coordinator site, which stores this information and associates it to the corresponding super-activity. The event is also notified to the global project manager.

The project manager of a local site involved in a global activity must select the human resources that will participate in that activity. When this is done, a "Global User Assigned" event is sent to the project management tool of the global site, to store this information at global level. The event is also notified to the global project manager.

5 Conclusion

In this paper we have presented the GENESIS approach to distributed project management. The definition of the GENESIS platform requirements for distribution, especially for the process modelling facilities, followed a strict interaction with the pilot users (the industrial partners) of the GENESIS project. The project started on September 2001 and ended on November 2003. The total effort was of 294 man-months and about 104,000 lines of code have been produced. Each GENESIS site is realized as a web application: the user interface and the coordination layer are realized using JSP (JavaServer Pages) and servlets (Tomcat being the web server), while the other components have been developed using the Java 2 Platform Standard Edition. The communication between the coordination layer and the different subsystems composing a GENESIS site is based on Java RMI, while the communication among the different sites is based on SOAP. The supporting database is based on MySQL Server.

The GENESIS platform has been evaluated by both the industrial and academic partners of the GENESIS consortium in different pilot projects. The code of the system has been delivered as open source software and can be downloaded from the SourceForge web site, <http://sourceforge.net>.

Acknowledgments

This work has been supported by the European Commission under Contract No. IST-2000-29380, Project GENESIS (Generalized ENvironment for procESs management in cooperatIve Software engineering, <http://www.ist-genesis.org>).

References

[1] C. K. Ames, S. C. Burleigh, and S. J. Mitchell. "WWWorkflow: World Wide Web based workflow", Proceedings of the 13th International Conference on System Sciences, pp. 397–404, 1997.

[2] V. Ambriola, R. Conradi, and A. Fuggetta. "Assessing Process-Centered Software Engineering Environments", ACM Transaction on Software Engineering and Methodology, vol. 6, no. 3, pp. 283–328, 1998.

[3] L. Aversano, A. Cimitile, A. De Lucia, S. Stefanucci, M. L. Villani. "GENESIS Workflow: Managing Distributed Software Processes", in Cimitile A., De Lucia A, and Gall H. (editors), Cooperative Methods and Tools for Distributed Software Processes, Franco Angeli, Italy, pp. 87–108, 2003.

[4] I. Z. Ben-Shaul and G. E. Kaiser. "Federating Process-Centered Environments: the Oz Experience", International Journal of Automated Software Engineering, 1997.

[5] Gregory Alan Bolcer and Richard N. Taylor. "Advanced workflow management technologies", Software Process Improvement and Practice, vol 4 n. 3, pp 125–171, 1998.

[6] D. Chan, and K.R.P.H. Leung. "A workflow Vista of the software Process", IEEE 8th International Workshop on Database and Expert Systems Applications (DEXA '97), 1997.

[7] G. Cugola. "Tolerating Deviations in Process Support Systems via Flexible Enactment of Process Models", IEEE Transactions on Software Engineering, vol. 24 no. 11, pp. 982–1001, 1998.

[8] G. Cugola, E. Di Nitto, and A. Fuggetta. "The JEDI Event-Based Infrastructure and Its Application to the Development of the OPSS WFMS", IEEE Transactions on Software Engineering, vol. 27, no. 9, pp. 827–850, 2001.

[9] J. Eder, E. Panagos. "Towards Distributed Workflow Process Management", AT&T Research Labs, 1999.

[10] J. C. Grundy, M. D. Apperley, J. G. Hosking and W. B. Mugridge. "A decentralized architecture for software process modelling and enactment", In IEEE Internet Computing, v 2 no 5, pp. 53–62, 1998.

[11]
D. Georgakopoulos, H. Hornick, and A. Sheth. "An Overview of Workflow Management: from Process Modelling to Workflow Automation Infrastructure" Distributed and Parallel Databases, vol. 3, no. 2, pp.119–153, 1995.

[12]
G. Q. Huang, J. Huang, and K. L. Mak. "Agent-based workflow management in collaborative product development on the internet", International Journal of Computer Aided Design, vol. 32, no. 2, pp. 133–114, 2000.

[13]
A. S. Hitomi, P. J. Kammer, G. A. Bolcer, R. N. Taylor. "Distributed Workflow using HTTP: Example using Software Prerequirements", Proceedings of the International Conference on Software Engineering ICSE'98, 1998.

[14]
B. Kötting, F. Maurer. "Approaching Software Support for Virtual Software Corporations", Proceedings of the International Conference on Software Engineering ICSE '99, Los Angeles, California, 1999.

[15]
F. Maurer, B. Dellen, F. Bendeck, S. Goldmann, H. Holz, B. Kötting, and M. Schaaf. "Merging Project Planning and Web-Enabled Dynamic Workflow for Software Development", IEEE Internet Computing, vol. 4 no.3, pp. 65–74, 2000.

[16]
F. Ranno and S. K. Shrivastava. "A Review of Distributed Workflow Management Systems", Proceedings of the international joint conference on Work activities coordination and collaboration, San Francisco, California, United States, 1999.

[17]
Workflow Management Coalition. "Workflow Management Coalition Interface 1: Process Definition Interchange Process Model", Document no. WFMC-TC-1016-P, 1999. Available at <http://www.aiim.org/wfmc/standards/docs/if19910v11.pdf>.

