A Communication Protocol for Distributed Process Management

Lerina Aversano, Aniello Cimitile, Andrea De Lucia

RCOST - Research Center On Software Technology
Department of Engineering, University of Sannio
Via Traiano, Palazzo ex-Poste – 82100, Benevento, Italy
(aversano/cimitile/delucia)@unisannio.it

Abstract

Large scale software development processes imply the coordination and cooperation of several sites with a large number of people and sub processes. We present an asynchronous communication protocol for distributed process management adopted within the GENESIS (Generalized ENvironment for procESs management in cooperatIve Software engineering) project. The GENESIS process management sub-subsystem enables distributed process modeling and enactment on different organizational sites through an event dispatching architecture.

1 Introduction

Workflow management is a rapidly growing research and development area of very high practical relevance in business applications and software development [CL97; GHS95; ACF98; WfM99]. However, most existing workflow management systems have a monolithic and centralized architecture and therefore are not adequate to cope with the requirements that large scale software development processes pose. Large scale software development processes imply the coordination and cooperation of several sites with a large number of people and sub processes [MDB00, BSK95, GAH98]. The support for distributed process management is a relevant problem for two reasons.

- distributed software processes may involve a large number of concurrent process instances which impose an adequate coordination support. Therefore, for sake of scalability and availability a software process needs to be distributed across multiple workflow engines running on the different sites involved;
- when a process spans multiple sites that generally work in a largely autonomous manner, it may be required that those parts of a process are under the responsibility of a local project manager that can organize the sub-process in a more appropriate way. Thus, the partitioning and distribution of a process may fall out naturally from the organizational decentralization.

In particular, the latter issue also poses problems concerned with the decentralized and autonomous modeling of distributed software processes. Most work on distributed process management focus on developing paradigms and architectures for the enactment of distributed processes and scarcely address decentralized process modeling [RS99, KM99, HK98, EP99, BSK97, GAH98, BDF96, C98, CDF01]. In most cases process modeling is a centralized activity and enactment of portions of the process is distributed on different workflow engines. In some cases, the central process model is collaboratively edited with the contribution of people on different sites [GAH98] and all sites have visibility of the overall process model. A different approach is used in OzWeb [BSK97] where process models are autonomously defined on the different sites and cooperates through specifically designed interfaces.

In this paper we present the GENESIS (Generalized ENvironment for procESs management in cooperatIve Software engineering) approach to distributed process modeling. GENESIS is an on-going research project aiming at designing and developing a non-invasive and open source system to support software engineering processes in a highly distributed environment [RG02]. The GENESIS process modeling language is hierarchical. The global process is modeled and enacted on the coordinator site (that is the technical leader of distributed software project [KM99]), while sub-processes can be autonomously modeled and executed on different organizational sites. The global process model can be collaboratively edited by the project managers of the different sites. This is particularly important to define the interfaces between the global process model and the coordinated sub-processes.

The paper is organized as follows. Section 2 discusses related work. Section 3 describes the overall architecture of GENESIS and the main features of the process modeling language, while Section 4 presents the asynchronous communication protocol for distributed process modeling and project management. Finally, Section 5 concludes and comments on future development.

2 Related work

In global and virtual enterprises, software processes consist of multiple sub-processes that may span organizational boundaries. 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 can not be fulfilled by traditional workflow systems [KM99, BT98]:

- support the distributed execution of a workflow;
- shared access to data,
- use of groupware tools
- support an easy transition from the old structure to the new one and within every project
- planning and change impact functionality.

Modern workflow management systems exploit the web as a mean to enable distributed access to the facilities provided by the workflow engine [ABM97, HHM00, BT98, MDB00]. However, most of these systems are still based on a client-server architecture and the problem of designing architectures for distributed process modeling and enactment of the process is still a research issue [RS99, KM99, HK98, EP99, BSK97, GAH98, BDF96, C98, CDF01, WfM99]. However, most approaches do not discuss issues concerned with distributed process modeling.
PROSYT is an artefact based PSEE [C98]. Each artefact produced during the process is an instance of some *artefact typ*e, which describes its internal structure and behaviour. All the routing in this model is based on the artefact and the operations on them. Boolean expressions are used to express the constraints under which operations are allowed to start. PROSYT also allows for distributed enactment facilitated by an event-based middleware [CDF01] (the same middleware is also used by the OPSS WfMS [CDF01]).
In [EP99] the authors propose an approach for the distributed execution that exploits the central role of an event notification service, 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. Moreover, time-related constructs for addressing the time aspects of process management are provided.
The Endevors project [HK98] 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 [KM99] propose an extension of MILOS [KM99] which focuses on the process support for virtual corporation by integrating the process modeling with project planning and enactment in distributed environments. They propose three different approaches for distributed process enactment:

- replicating the workflow engine (process model and data) on several servers and propagating a change to the data on one server to all the others;
- maintaining the process model on the central (coordinator) site and giving the possibility to the other servers to exchange the data they need for local process execution with the coordinator site;
- maintaining different portion of process model and data on different workflow engines which exchange data according to a peer-to-peer architecture.

The authors do not address the problem of decentralized process modeling. Grundy et al. [GAH98] focus on problems concerning the distribution in process modeling. The proposed system provides mechanisms for collaboratively editing process models both in a synchronous and 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 the model.
In the Ozweb environment the peer-to-peer paradigm for distribution is adopted [BSK95]. 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 corporation. Sub-processes executed on different sites are autonomously defines and only have to respect the interfaces defined at the global level. Moreover, the dept of the hierarchical process model is not limited at only two levels, thus allowing the possibility for a partner of the virtual corporation to have further sub-contractors.

3 GENESIS architecture for distributed process management

Traditional workflow management systems do not provide adequate support for the evolution of software organizations towards distributed virtual corporation models. The main open problem remains the systematic definition of distributed process models and their enactment across multiple sites using appropriate abstractions and mechanisms. GENESIS environment has been developed with the aim to cope these problems. The environment provide a special support for distributed scenario, from the modeling of a distributed process to its enactment.

Distributed projects in GENESIS are organized in a hierachical way. For example, a two level project will include a project coordinator site, managing the project at the global level and a number of local sites, managing specific project workpackages. The coordinator is in charge of modelling and executing the global process, while the local sites are in charge of modelling and executing the sub-processes concerning their workpackages. Therefore, each GENESIS site includes different components (see Figure 1):

- a workflow managament system to model and enact software processes;
- an artefact management system to store and retrieve the artefacts produced within a process;
- a resource management system to allocate resources, in particular human resources, to a project;
- an event engine to collect and dispatch events raised during process management, such as the termination of an activity or the production of an artifact, in particular between the coordinator and local sites;
- a notification and communication system to enable users of the environment to communicate and to send notifications about particular events of interest.

Distributed process modelling is made in an asynchronous way (see next section for the communication protocol). The global process model includes super-activities corresponding to workpackages that are associated to the sites where the actual work is performed. The local site independently creates the process model for the management of the workpackage. The only requirements are that a sub-process has been created when the corresponding super-activity in the global process model has to be enacted and the sub-process interface (in terms of input / output artefacts) is conform to that of the macro-activity (see Figure 2). Besides super-activities, a process model can also contains global activities, i.e. activities that can be collaboratively performed by workers distributed across different sites. The project manager of each site involved in a global activity is in charge of providing the needed human resources for the activity.

In GENESIS the process modeling 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. At both levels, our concern has been to create a language for defining processes in a way general enough to respect the single organization rules, in order to keep low the intrusiveness of the platform. In this respect, each activity (or work item) of the process model is essentially described by the artifacts that will produce, and freedom is left to the worker(s) to decide how to actually perform that activity.

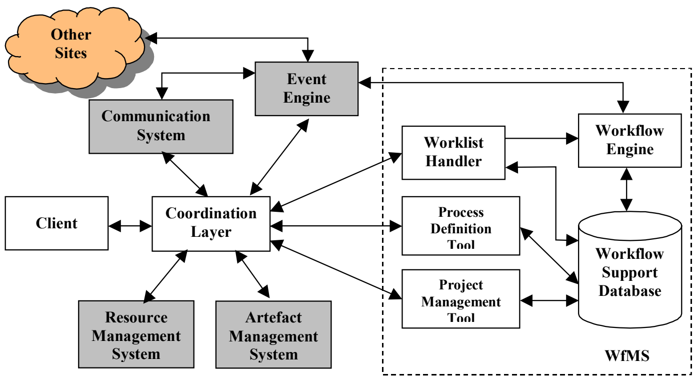

Figure 1 – GENESIS site architecture

GENESIS provides two process modelling levels. At the first level a process designer is able to create abstract process models, through the Process Definition Tool, according to specific organization standards. Abstract process models include descripiton 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 (artefact) flow between activities and are expressed according to the Event-Condition-Action paradigm. Abstract process models have to be customized for specific projects by specifying project data, such as actual people performing activities and actual artifacts. Project managers can use them as templates to create

concrete process models, through the Project Management Tool, for a specific project that can be enacted by the workflow engine. Different process instances may be created and enacted according to the same concrete process model.
Further details concerning the process modelling language are described in [AC03].

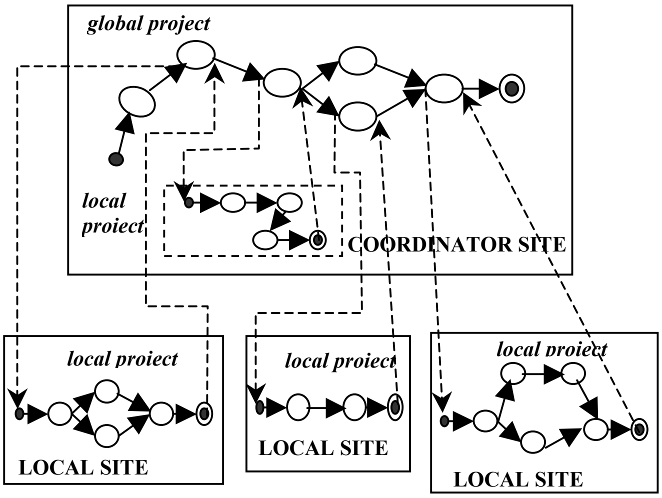

Figure 2 – Hierarchical project decomposition

4 Distributed process modeling

In the GENESIS environment, 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.
The protocol used for the communication is described in the following. We distinguish three main phases: the creation of the project on both 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 starting of a new project the resource manager of the global site creates a project using the resource management tool provided by the platform. This means that s/he selects the human resources allocated on the global site and the local sites participating to the project. A *"Global Project Creation"* event is sent to the involved sites and notified to the global project manager. The resource management tools that locally receive this event automatically store this information locally. Then, each resource manager of an involved local site decides the allocations of the human resources and the local project manager. The corresponding event is "*Local Project Creation*", which is sent to resource management tool of the coordinator site, to store this information at the global level. The event is also notified to the global and local 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[1], 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 manager.
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 this information and associates them locally to the project. The event is also notified to the project manager of the local site, as soon as s/he will be available.
For each global activity, the GPM can select for each site the number of required people and send this information together with the role associated to the global activity and the work modality (collaborative/single user) to the project manager of the local sites. A "Global Activity Creation" event is sent to each site involved in the global activity.
A concrete global process can start when it is completed, independently of the local process definition status (see next sub-section). Checks are made at enactment time to make sure that all super-activities and global activities have the needed resources allocated.

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). When the concrete local process is completed, 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

[1] It is worth noting that within a project different process models can be used for different purposes; this means that there is a 1 to many relations between projects and concrete process models.

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 to the 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 the 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 process modeling. The definition of the GENESIS platform requirements for distribution, especially for the process modeling facilities, followed a strict interaction with the pilot users (the industrial partners) of the GENESIS project [BC03]. At the starting of the project we collected detailed description of their work modalities on distributed processes and emerging needs. We considered all the problems express by them when organizing and conducting distributed processes and translated them in formal requirements for the implementation of the workflow management system of the GENESIS platform.
Currently, a single site implementation of GENESIS is already available, while the implementation of the features for distributed modeling and enactment is in progress. Each GENESIS site is realized as a web application: the user interface and the coordination layer are realized using JSP and servlets (Tomcat being the web server), while the other components are 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. The WfMS supporting database is based on MySQL Server.

ACKNOWLEDGMENTS

This work has been partially supported by the European Commission under Contract No. IST-2000-29380, Project GENESIS (GENERALIZED ENVIRONMENT FOR PROCESS MANAGEMENT IN COOPERATIVE SOFTWARE ENGINEER-ING) (http://www.ist-genesis.org).

References

[ABM97]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*, 1997, pp. 397–404.

[ACF98] V. Ambriola, R. Conradi, and A. Fuggetta, "Assessing Process-Centered Software Engineering Environments", *ACM Transaction on Software Engineering and Methodology*, vol. 6, no. 3, 1998, pp. 283-328.

[AC03] L. Aversano, A. Cimitile, A. De Lucia, S. Stefanucci, M. L. Villani, "Workflow Management in the GENESIS Environment", 2nd *Workshop on Cooperative Supports for Distributed Software Engineering Processes*, Benevento, Italy, 2003.

[BC03] D. Ballarini, M. Cadoli, M. Gaeta, T. Mancini, M. Mecella, P. Ritrovato and G. Santucci, "Modeling Real Requirements for Cooperative Software Development: A Case Study", 2nd *Workshop on Cooperative Supports for Distributed Software Engineering Processes*, Benevento, Italy, 2003.

[BDF96] S. Bandinelli, E. Di Nitto, and A. Fuggetta, "Supporting Cooperation in the SPADE-1 Environment", *IEEE Transactions on Software Engineering*, vol. 22, no. 12, 1996, pp. 841-865.

[BSK95] I. Z. Ben-Shaul and G.E. Kaiser, "A Paradigm for Decentralized Process Modeling and its Realization in the Oz Environment", *Proceedings of the International Conference on Software Engineering ICSE'95,* 1995.

[BSK97] I. Z. Ben-Shaul and G.E. Kaiser, "Federating Process-Centered Environments: the Oz Experience", *International Journal of Automated Software Engineering*, 1997.

[BT98] Gragory Alan Bolcer and Richard N. Taylor, "Advanced workflow management technologies", *Software Process Improvement and Practice*, vol 4 n. 3, pp 125-171, 1998.

[CL97] 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.

[C98]G. Cugola, "Tolerating Deviations in Process Support Systems via Flexible Enactment of Process Models", *IEEE Transactions on Software Engineering,* vol. 24 no. 11, 1998, pp. 982-1001,.

[CDF01] 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, 2001, pp. 827-850.

[EP99] J. Eder, E. Panagos, "Towards Distributed Workflow Process Management", AT&T Research Labs, 1999.

[GAH98]J. C. Grundy, M. D. Apperley, J. G. Hosking and W. B. Mugridge, "A decentralized architecture for software process modeling and enactment", *In IEEE Internet Computing*, v 2 no 5, 1998, p 53-62.

[GHS95] 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, 1995, p.119-153.

[HHM00] 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, 2000, pp. 133-144.

[HK98] Hitomi A. S., Kammer P. J., Bolcer G. A., Taylor R. N., “Distributed Workflow using HTTP: Example using Software Pre-requirements”, *Proceedings of the International Conference on Software Engineering ICSE’98,* 1998.

[KM99] 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.

[MDB00] 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, 2000, pp. 65-74.

[RG02] P. Ritrovato and M. Gaeta, “Generalised Environment for Process Management in Co-operative Software”, *Proceedings of the 26th Annual International Computer Software and Applications Conference, special section on the Workshop on Cooperative Supports for Distributed Software Engineering Processes*, Oxford, UK, IEEE Computer Society Press, 2002.

[RS99] 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.

[WfM99] Workflow Management Coalition, “Workflow Management Coalition Interface 1: Process Definition Interchange Process Model”, Document no. WFMC-TC-1016-P, 1999, accessible from http://www.aiim.org/wfmc/standards/docs/if19910v11.pdf.