allows for sound group coordination. The paper describes a mail based workflow engine that uses these aspects to enhance coordination. A 'distributed software development' (DSD) clerk is used to communicate information between members of the team. Reviews of modifications to the system are done anonymously but kept in a database. The paper also discusses some practical experiences and some problems that arose in implementation. A stand-alone application, DSD Client, was implemented but required installation on every member's machine. A solution to use Jini is provided to allow different kinds of programs to connect automatically.

Preliminary Results of an Industrial EPG Evaluation by L. Scott, R. Jeffery, U. Becker-Kornstaedt

This paper presented a case study on the usefulness of electronic process guides (EPGs). As companies define and/or modify their processes, it is important that the processes and their changes be communicated to all involved in the process. The usual and often ineffective method of communication is a paper-based memo. The Internet, however, provides an effective method of communication regarding processes within an organization. The web already provides distributed software teams with much assistance and these are discussed briefly by the authors. The focus, however, is on an evaluation of an electronic process guide. The paper looks at what the EPG is used for and how it is used, what the outcomes were, benefits and possible improvements. An EPG consists of a project navigation tree, a main page and an individual page for every artifact, activity, role and tool. A survey was taken to evaluate the outcome of the EPG and the results were encouraging from all sides but improvements were suggested. One primary use of the guide was to provide access to template documents for process steps. Evaluations of the presentation of the guide concerned the process rather than the actual presentation of the guide.

Approaching Negotiation Automation for Software Development Companies by Boris Kötting, Frank Maurer

In any company, better performance can be achieved by distributing tasks within the company efficiently. The same is true of globally operating software development teams. A problem arises though, when a manager cannot assign tasks according to the employee's skill set as the required skills are not available in her local team. In terms of software development, one must make many decisions regarding varying topics, thus negotiating tasks is complex. The paper discusses automating negotiation for agents in a distributed company. A virtual marketplace is created to enable negotiation between different companies. The paper provides four different strategies, a Yes/No offer, a sealed bid, an English auction and finally negotiation. To support automatic negotiation a utility function is used with a lower and an upper threshold. Values are determined based on user settings. The agent first checks the negotiation type, then determines the proper course of action. The paper notes a key point that the details of the utility function should remain hidden from the user. Finally, the possibility of multi-bids is discussed briefly.

Process Model Integration in Internet-based Virtual Software Corporations by Quentin Mair & Zsolt Haag

The Internet and increasing sizes of software development teams have changed the environments in which software is being developed, and virtual software companies (VSC) are a prime example. Tool support is required for process management and the desire to integrate and interoperate process models has defined the need for a VSC component process. Identifying the atomic processes performed within one organization helps to integrate varying processes between VSC members. A Process Specification Language defines a neutral language for process specification and assists in the exchange of process information. Finally, the Resource Description Framework provides interoperability between tools that exchange information over the Internet. The paper looks at four advantages in using RDF to define PSL and discusses five parts of the PSL declaration. A mapping of PSL/RDF to enactable process models is described and deficiencies such as homogeneity and lack of an industrial case study are discussed.

References

[1] First Workshop on "Software Engineering over the Internet". Web proceedings available at http://sern.ucalgary.ca/~maurer/ICSE98WS/ICSE98WS.html

[2] Second Workshop on "Software Engineering over the Internet". Web proceedings available at http://sern.ucalgary.ca/~maurer/ICSE99WS/ICSE99WS.html

[3] Third Workshop on "Software Engineering over the Internet". Web proceedings available at http://sern.ucalgary.ca/~maurer/icse2000ws/ICSE2000WS.htm

[4] Forth Workshop on "Software Engineering over the Internet". Web proceedings available at http://sern.ucalgary.ca/~maurer/icse2001ws/ICSE2001WS.html

Process Support for Distributed Team-based Software Development Workshop

Pierre F. Tiako
Inria-Irisa Lab, France
Pierre.Tiako@irisa.fr

Tim Lindquist
Arizona State University East, USA
Tim@asu.edu

Volker Gruhn
University of Dortmund, Germany
Gruhn@ls10.cs.uni-dortmund.de'

Abstract

This report summarizes the 2nd International Workshop on Process Support for Distributed Team-based Software Development held at the Sheraton World Resort of Orlando, Florida, on July 25, 2000 in conjunction with the Information Systems, Analysis and Synthesis (ISAS2000) International Conference. An overall twenty people attended the workshop consisting of seven technical presentations in two plenary sessions. In the following, we outline the presentations and subsequent discussions, which included modeling and distributing process component, evolution and change, web-based framework, consistency management, and reuse and interoperability.

The hardcopy of the papers selected for PDTSD'00 are published by the International Institute of Informatics and Systemics as a part of the "*Industrial Systems*" volume of ISAS 2000, ISBN 980-07-6695-2.

Workshop Goal

Due to globalization, it becomes necessary for big companies to organize their production in autonomous and geographically distributed environments. Resulting problems for the companies, which develop the software on Process-centered Software Engineering Environments (PSEEs) were in the scope of this forum.

The workshop sought participants to cover practical and theoretical issues of distributed team-based software development addressing the following key questions.

- How can we model and distribute process components, their internal and external structure and behavior, the services they can provide in a network?
- How can we manage evolution and change as part of the collaboration and the coordination of teams during the development?
- How to use the Web to organize the geographically distributed software development ?
- How can we manage the consistency, detect, prevent and resolve conflicts of multiples teams working in parallel?
- What kind of infrastructure can enable reuse, share and exchange the objects among geographically distributed teams?

Reflecting these goals, PDTSD'00 provided a forum where emerging approaches were presented, compared and discussed. It addressed basic issues, especially multiple facets of distributed software development. The workshop was organized in two technical sessions. In each session, four or five papers were presented. Each presentation was followed by a short plenary discussion.

Modeling and distributing process component

Oquendo [4] presented the p-SPACE architecture description language and its architecture-driven approach for designing and generating component-based software process models. p-SPACE addresses support for compositionally of architectural elements and the incremental synthesis of process model components. The presentation by Lindquist [10] highlights a system of cooperating distributed user interfaces that jointly manipulate the process components. He reports on applying JINI and JavaSpaces to achieve distributed interoperable process components. He also presents the architecture and realization of a prototype system providing plug-in discovery of available process services. Maheshwari [12] presented a distributed software project management information system, the DSPMtool. It consists of the tools and techniques used to gather, analyze, integrate, and disseminate the outputs of various software processes in order to facilitate teamwork.

Evolution and Change

Ajila [2] presented an approach to manage the change as part of the collaboration and the coordination of teams during the development. He is based on the fact that software development is opment. He is based on the fact that software development is a distributed activity involving different people in diverse geographical locations. These people have different roles, backgrounds, and tastes. Collaboration and coordination are needed for team-based software development. He proposed a case study based on an integrated software development environment that includes change management tools. He supported the distribution of change related information by simple tools. The p-SPACE presented by *Oquendo* [4] supports the evolution of software processes through regeneration of components according to changes in the process architecture in which they operate. Thereby, generated components remain compliant as their architecture evolves *Jorgensen* [8] proposed a framework with four activities, process model improvement, reuse, enactment and harvesting to generate a set of requirements for process model evolution.

Web-Based Framework

De Man [5] described the evolutionary development of a process-centered project support environment. He recalled the main motivators and how they influenced implementation decisions, in particular the choice of a lightweight approach using standard web technology. He presented an early experience and gave suggestions on how to increase and accelerate the acceptance of its propositions in a globally distributed organization. The paper by *Maidantchik* [11] presents a software process management model to a worldwide collaboration. It proposes applying a unique model of integrated processes, which is used to support their continuos improvement. The process model proposed is specialized according to each software group and instantiated for each project. A Web tool supports the enactment of process instances. *Belkhatir* [13] reported their experience in developing a temporal process for web multimedia applications over the Web. Systems, which run such applications can be centralized or distributed. He identified the principal strengths and weaknesses of both multimedia application and process sensitive systems as software systems and workflow management systems. He makes some proposals to improve web multimedia applications. The web was chosen because of its generalization, low-cost and easy to use principle.

Consistency Management

Finkelstein and *Smolko* [6] gave an account of architecture for management of consistency relations between distributed documents. They proposed a collaborative framework of interacting software agents, capable of concurrent execution of consistency checks and architectural performance evaluation and scalability. The proposition demonstrates the advantages of using of this system in a domain of software engineering documents. To prevent consistency, *Hoen* [7] defended the idea for coordinating the efforts of multiple teams working in parallel on a model. A major part of this effort is to resolve conflicts, which are only detected when the work of the separate teams is integrated. He presented how a model can be cut into distinct packages where in parallel each of these packages is locally modified by just one of the teams. Integration of the modified packages is straightforward as he only allow local changes to a package, i.e. changes that do not propagate beyond the package and that do not cause conflicts during integration. Additionally, he showed how the package structure of a model and the teams working on the packages could be (temporarily) adapted to manage the need for non-local changes.

Reuse and Interoperability

Belkhatir [3] has suggested an adaptation of the observer pattern structure by Gang of four to support management and prevent to possible deviation of a team. He showed how we adapt the observer pattern to support monitoring of processes. His use of the observer pattern results in an architecture that consists of several components. He considered an observer as a management tool which reports back to process administrators with a clear picture of the overall process state. When applied, the Observers keep multiple views of the process state in synchronization with the process state server. Observers can be deployed during the enactment of the process and can operate as a back end tool weakly coupled to the process engine enacting the process. In order to coordinate and integrate the process observation driven and the event-driven process execution driven, he proposed an event server, which captures occurrences of events that arise during process execution.

Models of software development processes are used for assessing and improving work practice, and for supporting the enactment of work in software projects. Integrating process enactment and improvement would enable a down-to-earth approach to knowledge management and learning anchored in real practice. Jorgensen [8] presented a reference model for such integrated process knowledge management. The framework has activities, which generate a set of requirements for process model reuse. He evaluated the mechanisms for process model inheritance based on these requirements, illustrating the applicability of the framework. He defended the idea that inheritance must be combined with approaches like components, projection, parameterization and patterns to fulfil all requirements. *Tiako* [9] argued that tools are not always compatible with the different objects that must be shared or exchanged among geographically distributed teams involved in a large project. It is still important to provide an infrastructure that enables to share or exchange object among teams. He described the basic concepts of such an infrastructure.

Conclusion

PDTSD'00 contained an abundance of insightful and meaningful communication amongst the researchers in the field. Attendees came and they talked. In this sense, the workshop completely fulfilled its goals.

Paper References

[1] Proceeding of the International Conference on Information System, Analysis and Synthesis - ISAS 2000 - "Industrial Systems" volume, Orlando, FL, July 23-26, 2000, 760 Pages. Ed. Sanchez, B., R. Hammel, M. Soriano and P. Tiako.

[2] Ajila, S. Change Management in a Team-based Distributed Software Development Environment: A Case Study of Basic Supports using WWW, pp. 688-694, In [1].
Keywords: change management, distributed team-based development, WWW, and software evolution.

[3] Belkhatir, N., N. Mihoubi. Adapting the Observer Pattern for Process Management, pp. 694-699. In [1].
Keywords: process model, process sensitive systems, OO framework, logging, process management, observer pattern, design pattern, process state, container, events, triggers.

[4] Chaudet, C., K. Megzari, F. Oquendo. A Formal Architecture-Driven Approach for Designing and Generating Component-Based Software Process Models, pp. 700-706. In [1].
Keywords: architecture-driven software process engineering, architecture description language, architecture-based synthesis of process models, process components.

[5] De Man, J. A lightweight process-centered project support environment: motivation, implementation and experience, pp. 708-714. In [1].
Keywords: process-centered support environment, web technology, capability maturity model, process model, process tailoring.

[6] Finkelstein, A., D. Smolko. Software Agent Architecture for Consistency Management in Distributed Documents, pp. 715-719. In [1].
Keywords: consistency management, XML, software agents, co-operation, mobility, distributed architecture.

[7] 't Hoen, P.J., and M.H. ter Beek. A Conflict-Free Strategy for Team-Based Model Development, pp. 720-725. In [1].
Keywords: conflict strategy , package change, team automata.

[8] Jørgensen, H. D. Software Process Model Reuse and Learning, pp. 726-731. In [1].
Keywords: process model reuse, process model inheritance, knowledge management, process model evolution.

[9] Tiako, P. F., and G. Kouamou. Tool Integration in Distributed Software Development Environments, pp.732-737. In [1].
Keywords: interoperability, CORBA, heterogeneity and distribution, software tool.

[10] Lindquist, T., K. Gary. Experience with Distributed Process Components on Jini and JavaSpaces, pp. 738-743. In [1].
Keywords: process component, software process, distributed applications.

[11] Maidantchik, C., A. R. C. Rocha and G. B. Xexéo. Process Management in a Worldwide Collaboration, pp.744-748. In [1].
Keywords: software process, maturity model, worldwide software development, process management and improvement.

[12] Maheshwari, P., and R. Surjaputra. DSPMtool: A Tool for Distributed Project Management, pp. 749-754. In [1].
Keywords: project management, client-server, software engineering, configuration management.

[13] Villanova, M., N. Belkhatir, N., H. Martin. A Temporal Process Model for Web Multimedia Applications, pp.755-760. In [1].
Keywords: multimedia, process sensitive system, OO framework, web based systems, temporal relationships.

4th ICSE Workshop on Component-Based Software Engineering:

Component Certification and System Prediction

Ivica Crnkovic[1], Heinz Schmidt[2], Judith Stafford[3], Kurt Wallnau[3]
[1]Mälardalen University, Department of Computer Engineering, Sweden, ivica.crnkovic@mdh.se
[2]Monash University, Australia, Heinz.Schmidt@csse.monash.edu.au
[3]Software Engineering Institute, Carnegie Mellon University, USA, {jas, kcw}@sei.cmu.edu

Abstract

This paper gives a short overview of the 4th ICSE Workshop on Component-based Software Engineering. The workshop brought together researchers and practitioners from three communities: component technology, software architecture, and software certification. The goal of the workshop was to find a common understanding and to discuss the topics related to the component composition. The workshop was divided in eight sessions held in sequence, starting with invited talks and ended with a final discussion. A model problem, to be used for further research and work in future workshops, was discussed and later selected. The paper gives a comprehensive summary of the sessions and plans for future work.