Project CoBE: Supporting the Collaboration of Planning Processes through a Component-based Peer-to-Peer Architecture

Sascha Alda, Armin B. Cremers
University of Bonn, Germany

This article summarizes the research findings of the project CoBE. The purpose of CoBE is to support collaborative planning processes in huge civil and building engineering projects. The main contribution of CoBE is the DeEvolve platform, which is a component-based peer-to-peer architecture. By means of this architecture, members of a cooperation are capable of sharing partial data models, services, and hardware resources with other members of the same cooperation. The incorporation of the component technology allows users to integrate remote data and services with other local applications (components) in a flexible way. The second part of CoBE is a decentral Awareness Framework. This framework supports the coordination of working activities especially on distributed partial planning models. Besides presenting the concepts of both the DeEvolve and the Awareness Framework, we also stress on two possible application scenarios, in which the findings of CoBE could have been applied.

1 Introduction

A plethora of different engineers working collaboratively in virtual projects teams, nowadays characterize recent planning processes in complex civil and building engineering projects. As a typical characteristic of virtual teams (hereafter indicated as *networked cooperations*), projects members reside in different locations such as in different cities or even different countries and work temporarily asynchronous due to the time shift of different countries. Networked Cooperations exhibit highly dynamic and loosely coupled structures, since members (e.g. structural designers, engineers, technicans) join and leave the cooperation depending on their assigned responsibility.

The collaboration within networked cooperations is typically enhanced by software or *groupware systems*, such as email, chat, file sharing, or service sharing systems. Most of such systems are, today, based on the centralized client-server architecture model having one dedicated server and potentially many clients. This architecture model certainly fits well for static organizations being of a (relatively) fixed size and characterized by a centralized place of control (e.g. a chief office). However, for loosely coupled co-operations with permanently changing conditions, as it is typical for complex structural design projects, this model seems not appropriate. Another challenge one has to incorporate for the deployment of a groupware systems constitutes the *heterogeneous* environment of these cooperations, since protagonists due to work with a plethora of different software systems, data models and formats, and operating systems.

In the course of our research project CoBE[1] we have analysed new architecture models for groupware systems in order to enhance the collaboration of networked cooperations in planning processes. Our main contribution is a component-based peer-to-peer software architecture called DEEVOLVE. This architecture allows the flexible

[1] This project is funded in the course of the priority program 1103 „Networked-based co-operative planning processes in structural engineering.

deployment of groupware (and software systems in general) supporting the collaborative planning of distributed planning models. According to the underlying peer-to-peer model, each member is thereby capable not only of providing, but also of consuming individual partial planning models, auxiliary services, and hardware resources that can be shared with other members of the same cooperation. On top of the DEEVOLVE architecture, we have implemented the COBE AWARENESS FRAMEWORK supporting the coordination of working activities especially on distributed partial planning models by making planning activities explicit for other cooperation members.

The rest of this paper is organized as follows: section 2 presents some technical background information for both the DEEVOLVE architecture as well as the Awareness Framework. The third chapter then stresses on two different application fields, in which the findings of our project could have been successfully adopted. In both chapters, the reader is also referred to further literature of the authors, in which the respective topics have been elaborated in more detail.

2 Technical Background of CoBE

2.1 DEEVOLVE – A component-based peer-to-peer architecture

The DEEVOLVE architecture [Alda *et al.*, 2002] [Alda and Cremers, 2003] [Alda and Cremers, 2004] follows the *peer-to-peer architecture model*. This architecture model is a novel abstraction for developing software aimed to support virtual organizations. Following this style, a peer-to-peer architecture constitutes a distributed architecture that consists of uniform clients or so-called *peers*. Peers are capable not only of consuming, but also of providing computer resources like data, legacy applications, proprietary software solutions, simple routines, or even hardware resources. These resources are encapsulated by *peer services*. In contrast to other service-oriented architectures, peer-to-peer architectures assume an unstable and dynamic topology as an important constraint. That is because peers are solely responsible to affiliate to a network. Beyond the possibility of direct resource sharing, peer-to-peer architectures enable single peers to organize into so-called *peer groups*. These self-governed communities can share, collaborate, and communicate in their own private web. The purpose is to subdivide peers into groups according to common interests or knowledge independent from any given organizational or network boundaries.

DEEVOLVE also incorporates the component technology as a fundamental technology. According to the component technology, peer services are made up by the composition of single software components. A component model called FLEXIBEAN does prescribe the valid interaction primitives for both the local interaction within a service and the remote interaction among distributed services. Peer services can be made available (published) to other peers by means of advertisements. Each service can be assigned to at least one or more group affiliations, in order to restrict the access to a service for authorized group users. Users of other remote peers are able to discover and use these services. Additionally, we provide a composition language called PeerCAT [Alda and Cremers, 2004], enabling users to declare the composition of different peer services towards individual, higher-level applications. As a self-adaptable architecture, DEEVOLVE is capable of detecting and resolving unanticipated exceptions such as the failure or unavailability of peers. The handling of exceptions is done in strong interaction with end-users: an end-user is, for instance, able to decide which routine is to be executed for resolving an occurred exception. DEEVOLVE is accompanied by a couple of auxiliary tools not only for the discovery, advertisement, and composition of services, but also exception handling, as well as for the management of groups. Most of these auxiliary services have been adopted from the JXTA framework by Sun, which constitutes the state-of-the-art framework for the development of peer-to-peer architectures.

2.2 CoBE Awareness Framework

On top of the DEEVOLVE platform, we have developed the CoBE AWARENESS FRAMEWORK [Alda and Cremers, 2003], which realizes a decentral awareness model to coordinate the working activities within a project. This framework can be used by other applications to notify peers (that is, cooperation members) about changes within the application. The purpose of this model is to make the interactions between peer services explicit for end-users. End-users are thus accomplished to derive further actions based on the sole awareness of precedent activities stemming from other users who, for instance, belong to the same (project) group. Users can either be notified directly on the screen or, if they are currently unavailable, asynchronously via email. In order to avoid any violation of privacy of individual cooperation members, each member is capable of defining so-called *filter agents*. The purpose of these agents is to pre-select events of activities that should be made explicit for other users. Also, filter agents allow the selection of non-relevant events stemming from other members.

3 Application Scenarios

In cooperation with projects of the University of Bochum and Darmstadt, both also belonging to the priority program 1103, we have identified two application scenarios, in which we could apply and evaluate the findings of the CoBE project. In collaboration with the Darmstadt project, we demonstrated, how the CoBE AWARENESS FRAMEWORK could be taken for the collaborative planning of fire prevention concepts in building design [Alda *et al.* 2003]. The results of the cooperation with the University of Bochum is summarized in the following subsection.

3.1 Support of Collaborative Structural Design Processes through the Integration of Peer-to-Peer and MultiAgent Architectures

In cooperation with the project from the University of Bochum, we have analysed the pros and cons of the application of both the peer-to-peer and the multiagent architecture style for the usage the field of collaborative structural design [Alda *et al.*, 2004]. The potential of the multiagent architectural style for supporting planning processes has been studied in-depth by the University of Bochum for the last three years. Apparently, the agent system ACOS, which was realized within this project, exhibits some conceptual advantages compared to the CoBE platform. Basically, *autonomous computing* is in fact one of the main benefits to fall back on the agent-oriented ACOS architecture. According to ACOS, agents are capable of perceiving their environment and, based on the acquired information, capable of executing actions back to the environment autonomously. Compared to the rather end-user focused approach arrogated by CoBE, the surplus value of this approach can be seen in the guaranteed and immediate execution of actions. If, for instance, an engineer has changed a part of an overall structural design model, then an agent could react instantaneously to check the consistency of this model in dependency to all other parts. However, the agent-based architecture model also exhibits some weaknesses. In contrast to components, agents are rather inappropriate for the integration and flexible composition of heterogeneous software. This is due to the communication overhead that may result if several agents interact. Unlike components, which feature an intuitive and easy way for interaction, agent interaction is based on an ontology-based communication.

In order to eliminate the weaknesses exposed by both the CoBE as well as the ACOS architecture, we have conceptualised and implemented an integrated platform termed MAS-P2P [Alda *et al.*, 2004]. With respect to the MAS-P2P integration, not only (human) members, but also agents are able to subscribe to the awareness framework for perceiving activities. These agents act on the behalf of a human member by reacting immediately and autonomously on distinct events. A special *consistency agent* is thereby able to perceive activities concerning the modification of a local model that has been carried out by an individual member. This agent type is then responsible to check, whether or not the consistency of the overall model has been violated.

4 References

[Alda *et al.*, 2004]
Alda, S. Cremers, A.B., Bilek, J., and Hartmann, D.: "Support of Collaborative Structural Design Processes through the Integration of Peer-to-Peer and Multiagent Architectures", in: *Proceedings of the 10th International Conference on Computing in Civil and Building Engineering (ICCCBE-X).* Weimar, Germany.

[Alda and Cremers, 2004]
Alda, S. and Cremers, A.B.: "Strategies for Component-based Self-Adaptability Model in Peer-to-Peer Architectures", in: *Proceedings of the 4th International Symposium on Component-based Software Engineering (CBSE7).* Edinburgh, Scotland. May, 2004.

[Alda and Cremers, 2003]
Alda, S., Cremers, A.B.: "Collaborative Support for Planning Processes through Component-Based Peer Services", accepted for: *10th International Conference for Concurrent Engineering (CE 2003)*. Madeira, Portugal.

[Alda *et al.*, 2003]
Alda, S., Cremers, A.B., Meißner, U.F., Rüppel, U., Theiß, M., "Wahrnehmung und Verarbeitung von Ereignissen bei der verteilten Planung im baulichen Brandschutz", in: *Proceeding des 16. Internationalen Kolloquiums über Anwendungen der Informatik und Mathematik in Architektur und Bauwesen. IKM 2003*. Weimar, Germany, Juni 2003.

[Alda *et al.*, 2002]
Alda, S., Radetzki, U., Bergmann, A., Cremers, A.B., "A Component-Based and Adaptable Platform for Networked Cooperations in Civil and Builidng Engineering", in: *Proceedings of the 9th International Conference on Computing in Civil and Building Engineering (ICCCBE-IX).* Taipei, Taiwan, April 2002.