The Remote Control Approach – An Architecture for Tutoring in Pre-existing Collaborative Applications

Andreas Harrer, Nils Malzahn, and Benedikt Roth

Collide Research Group, University Duisburg-Essen, 47048 Duisburg
{harrer, malzahn.roth}@collide.info

Abstract. In this paper we present an architecture for the integration of tutoring and process scaffolds into existing collaborative applications. The architecture allows to combine existing research results concerning collaborative processes and their formalization, and existing and tested collaborative learning environments. The architecture allows controlling the learning environments either by a human or a pedagogic agent. Both types of tutors are using the same set of primitives – either via an intuitive user interface or a slim Java interface.

1 Structuring and Scaffolding Collaboration

Collaboration has become an important factor in learning activities, especially in disciplines that require substantial phases of working in teams, such as computer science, communication sciences etc. This can be seen in the emergence of the research field Computer-Supported Collaborative Learning (CSCL) in the last decade. Yet, just reducing the computer-based support to providing the suitable technological means to communicate is most often insufficient to promote the collaborative learning activity: Studies, like [1] showed, that collaboration does not happen effectively in every situation just by initiating the collaborative situation.

Scaffolds [2] or collaboration scripts are means to structure the learning activity and support the learners in organizing their activities or acquiring the skills to collaborate effectively. Thus their use in computer-based learning support environments (LSE) is a major topic of recent research in the CSCL community ([3];[4]).

Interestingly a parallel discussion occurs also in a field of computer-supported learning that has evolved independently of CSCL, the discipline of Intelligent Tutoring Systems (ITS): The support of the learners by the system to promote them in the learning process is often called tutoring or interventions.

It is obvious that the expertise and experiences of these two fields should be combined in collaborative computer-supported learning activities. One of the grand challenges for the shared interest between the communities will be the representation and implementation of scaffolds respectively tutoring for collaborative scenarios.. This article will present our approach of combining aspects from CSCL, pedagogical design, and ITS in an integrated architecture for supporting collaborative learning activities.

Up to now complex learning support environments and explicit scaffolding/tutoring models are largely unrelated and co-exist, but do not co-operate. On the one hand LSEs, such as WISE, CoLab, or Belvedere, either have a specific ("hard-wired") process model embedded or do not have an explicit learning process model at

M. Ikeda, K. Ashley, and T.-W. Chan (Eds.): ITS 2006, LNCS 4053, pp. 760 – 762, 2006.
© Springer-Verlag Berlin Heidelberg 2006

all. On the other hand environments that use explicit process models for supporting the learning process typically fall short in either re-usability or expressiveness of the process model. Most systems using a formal model define their own proprietary model for the learning process which is not understandable and thus re-usable by other applications. Systems that have explicit mechanisms for structuring activities usually tend to have a very narrow focus, such as sequencing the presentation of learning material in web-based hypertext systems or intervening on the first deviation from an "ideal" learning path. There are very few approaches that explicitly try to scaffold collaboration with adaptive approaches: coming from the ITS area, like the "Collaborative Tutor" approach and the GridCOLE approach from the CSCL field.

IMS Learning Design can be considered as a formal approach with explicit representation of both the models and the operational semantics. We assume that the formal character of IMS LD can also be utilized to scaffold and apply tutoring support for pre-existing LSEs. The availability of learning design engines (LDE), such as CopperCore , could provide explicit process support without having to implement a process model from scratch for each individual environment, if we can meet the challenge of integrating pre-existing LSEs and LDEs in a flexible, interoperable architecture. The idea is to combine the flexibility of learning scripts, which can be adapted to different learning groups and tasks, with the often task-oriented and domain specific ITS systems. Given this it will be possible to use one learning flow for more than one learning environment at the same time. That means the script (agent, tutor) can be used for arbitrary (within certain limits) collaborative learning environments, enabling students using different learning environments to collaborate with each other.

2 A Flexible Architecture for Tutoring in Collaborative Settings

We propose an approach that aims at a clear separation of the learning design engine together with the specification and implementation of the learning flow (as LD documents) and the collaborative learning environments. In this proposal we assume that the learners interact exclusively with the LSE without having to know anything about being "scripted" or "scaffolded" by the LDE respectively the LD document. In our approach the LDE is used as a process regulation facility that interacts with the LSE using a common generic vocabulary of communication primitives. This has the advantage, that the LD document can be used with a variety of different LSEs without any changes to the document.

For the concrete realization of our approach we defined an architecture that brings together LSEs and LDEs without having to make substantial changes in either of the two components: the schematic overview of the architecture can be found in figure 1 and the components introduced have the following function:

- Engine Extension (CopperCore Extension): this component extends the event propagation mechanism of the learning design engine so that events are sent to the LSE to remotely control the learning process via the Remote Control Component.
- Remote Control Component: this component maps events coming from the LDE to one or more communication primitives, that build the vocabulary for remotely controlling learning support environments

- LSE Remote API (Translator): this interface accepts communication primitives that have been defined for a variety of different LSEs and maps these primitives to the specific functionality available in the concrete LSE.

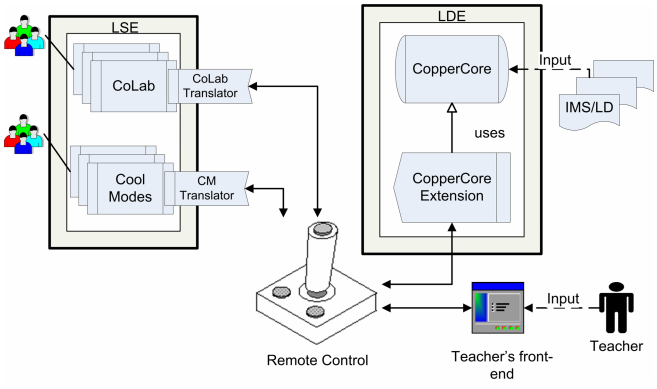

Fig. 1. Remote control architecture for interaction between LDE and LSE

An interesting feature of this architecture is, that besides our main purpose - the realization of collaboration scaffolds in pre-existing learning support environments - the remote control can be used by a variety of different actors such as intelligent tutors or human teachers.

Since the already present systems shall not be rewritten we decided to use a loosely coupled approach that allows to be adjusted for different learning support environments and different scaffolding agents to be applied. The proposed architecture has been implemented using the Cool Modes LSE and Copper Core Engine (LDE).

References

[1] Weinberger, A.:Scripts for Computer-Supported Collaborative Learning Effects of social and epistemic cooperation scripts on collaborative knowledge construction. PhD Thesis, Ludwig-Maximillans-Universität München.. (2003).

[2] Quintana, C., Krajcik, Joseph, Soloway, Elliot: A Case Study to Distill Structural Scaffolding Guidelines for Scaffolded Software CHI (2002).

[3] Weinberger, A., Fischer, F. Häkkinen, P., Dillenbourg,, P., Harrer, A.: Computer-supported scripting of interaction in collaborative learning environments. Workshop at the Conference on Computer Supported Collaborative Learning Conference (CSCL-05). Taipei, Taiwan (2005).

[4] Harrer, A., Hernandez Leo D. Dimitriadis, Y.: Languages for modeling of collaborative learning processes: Formalization, Practical uses and tools. Workshop at CSCL-05. Taipei, Taiwan (2005).

[5] Koper, R., Tattersall, C. (ed): Learning Design: A Handbook on Modelling and Delivering Networked Education and Training Springer Verlag (2005).