Towards Collaborative Learning via Shared Artefacts over the Grid

Cornelia Boldyreff, Phyo Kyaw, Janet Lavery, David Nutter, and Stephen Rank
Research Institute for Software Evolution, University of Durham,
Department of Computer Science, Science Laboratories,
South Road, Durham, DH1 3LE, UK
cornelia.boldyreff@durham.ac.uk

Abstract

The Web is the most pervasive collaborative technology in widespread use today; and its use to support eLearning has been highly successful. There are many web-based Virtual Learning Environments such as WebCT, FirstClass, and BlackBoard as well as associated web-based Managed Learning Environments. In the future, the Grid promises to provide an extremely powerful infrastructure allowing both learners and teachers to collaborate in various learning contexts and to share learning materials, learning processes, learning systems, and experiences. This position paper addresses the role of support for sharing artefacts in distributed systems such as the Grid. An analogy is made between collaborative software development and collaborative learning with the goal of gaining insights into the requisite support for artefact sharing within the eLearning community.

Keywords: collaboration, shared artefacts, grid, eLearning.

1. INTRODUCTION

Five major classes of grid applications identified by Foster and Kesselman [1] are as follows: distributed supercomputing, high throughput, on demand, data intensive, and collaborative computing. Any community intending to make effective use of the Grid must collaborate; and the Grid itself has enormous potential to support such collaboration. One important aspect of collaboration is artefact sharing. Often the asynchronous exchange of information via shared documents augments synchronous communication either in face-to-face meetings or through video conferencing.

The famous four by four matrix classification of groupware: same place/same time, same place/different time, different place/same time, and different place/different time, originally employed by Johansen, cited in [2] provides a overview of potential areas of applying groupware in collaborative learning in the grid context. These applications range from distance learning environments with video conferencing, computer based classroom systems for both collocated students and distance learners, to shared development and management of learning materials, processes, and experience reports. The latter application can be summarised as shared artefact development and management. The remainder of this paper will focus on support for artefact sharing.

Within the Genesis project [3], an Open Source Component Artefact Repository, OSCAR, [4] is being developed to support artefact management in general and more specifically collaborative software development via artefact sharing. In the following sections, both collaborative software engineering and collaborative learning are described, and an analogy is made between collaborative software development and collaborative learning and their respective associated activities. On the basis of this analogy, we argue that the support for artefact sharing developed within the Genesis project has the potential with suitable adaptation to be applicable to those engaged in collaborative learning and developing and managing shared artefacts. Artefacts in this context could be lesson plans, lesson content, student records, or other forms of educational artefacts.

2. COLLABORATIVE SOFTWARE DEVELOPMENT

The increasing globalisation of software development has led to the recognition of that large software systems are rarely the work of a single individual. During development of a software system, and throughout its evolution, a number of individuals will be involved. In many cases, the individuals concerned may come from distributed locations, for example, divisions within a company or from various co-operating companies working together in a single project. The team may develop a large software system from its inception. Certainly over its lifetime, a large software system will involve more than one person in its subsequent evolution. In this context, the software system constitutes a collection of shared artefacts; and it is highly likely that a number of associated shared artefacts will also result such as requirements documents, designs, program code, user manuals, test plans, and various other work products and forms of project documentation. Thus, in this setting, the sharing of artefacts and their management throughout the lifetime of the system is highly desirable to ensure continuity of knowledge within the team. During the lifetime of a system, the team associated with its development, deployment, and evolution is also subject to change as members leave and new members join. Having an established repository of shared artefacts is one means to easing the integration of new team members.

Equally important is the recognition that the processes whereby software systems are developed and maintained can be defined and refined as the basis of software process improvement within the software industry [5]. Furthermore defined processes may be enforced through process enactment systems, such as workflow management systems [6]. Defined processes that have proven successful are potentially reusable in the future, whereas unsuccessful processes may refined in the light of experience gained and may be reused in other projects. Such process definitions form another type of shared artefact and their refinement and reuse are often collaborative activities facilitated by storing them in an artefact management system, such as OSCAR, being developed by the Genesis project.

The Genesis project is focused on the development of a generalised environment for process management in cooperative software engineering [3]. The environment under development includes global project management as well as local site based workflow management to support both global and local work processes. A key system underlying these process management systems is the artefact management system, OSCAR. Additional services under development include resource management, event notification, metrics, and agent-based communications. OSCAR has been designed to provide generic support for artefact management. To support collaborative software engineering, the following specific types of artefacts have been pre-defined with XML document type definitions: software, documentation, software tool/service, process element, process instance, actor/user, and legacy [7].

At first it may seem odd to record details of people as actor (synonymous with user) in OSCAR. The rationale for this is to support access controls, but also to record historical data such as authorship of an artefact and membership of a project. In maintaining a software system, it is helpful to know the processes by which it has been produced and have access to the associated work products and their respective authors. In addition, recorded profiles of the software development personnel, both engineers and managers, are frequently used in large multi-national software companies in the selection of software project teams. Such records, profiles associated with actors, are another type of shared artefact that supports project managers in the selection of project teams. Where teams from previous projects are also recorded, these teams may also be potentially reusable. There are privacy issues that need to be considered with respect to recording personal data and these may be dependent on obtaining the consent of the individuals concerned.

In the software industry, much work has been done to standardise the activities of Software Engineering and their associated work products through professional organisations such as the IEEE [8]. More generally, the focus on software process improvement has resulted in the Capability Maturity Model [9] and more recently the SPICE standard [10]. Orthogonal considerations of quality through the development of the ISO 9000 standards family [11] have also been influential in the software industry. Thus, distilled industrial best practice is available for software organisations to use in the development of their own process models. In addition, more specific development approaches such as Rational Unified Process [12], RUP, are available with associated tool support. However, to facilitate collaboration across projects over time such tools require integration with an artefact repository, so that artefacts have the possibility of persistence and can be made more generally available to potential reusers.

3. COLLABORATIVE LEARNING

In the education domain, comparable standardisation is not so advanced internationally, nor so widely adopted although there are national efforts such as the Teaching Quality Assessment initiative in UK universities [13]. However, there are long and established traditions of learning and much literature describing educational processes, student-centred learning, and learning resources. An extensive literature survey can be found in Koper's report [14].

There are some standards under development that mirror the SPICE standard with respect to the introduction standard terminology and models for key concepts and processes in education. One standard recently proposed the Open University of the Netherlands defines the Educational modelling language [14], EML, an XML based notation to enable the codification of educational processes and typed learning objects within learning environments. More recent proposals have addressed the standardisation of collaborative learning systems [15]. A related proposed standard being developed by educational software producers in collaboration with user communities in education is the IMS Enterprise addressing exchange of data within managed learning environments and other systems within learning organisations such as virtual learning environments, student records, library management, and human resources [16].

These various education standards provide the basis for systematically describing educational processes, learning materials, student profiles, and educational outcomes in such a form that they could be recorded and stored in an artefact management system while under development and for collaborative reuse in the future by other educators and students. It may be that many educationalists are willing to contribute learning artefacts as open source. It is notable that EML is in the public domain.

4. ANALOGY BETWEEN COLLABORATIVE SOFTWARE DEVELOPMENT AND COLLABORATIVE LEARNING

If we compare the status of process modelling and standardisation with the two domains of software engineering and education, there are many points of agreement. Both are concerned with assuring the quality of their processes and products. In both domains, computer-based systems support the key activities. Both rely on computer-based representations of key work products. Many of the management processes in both domains are heavily supported by computer systems. In addition, in both domains, individuals as well as groups are key participants; and in neither case, is it sensible to speak of any activity without reference to others. The individual system designer relies on concepts established by others; just as an individual learner consults literature written by others; and in any case, both work within well established domains of theory and practice.

In both domains, collaboration is well established, as is the practice of computer-supported co-operative working. To ensure results of such work can be easily shared across each community, open source developers, software company project teams, students and teachers within an institution of learning or working in an international learning network, support for artefact sharing in both domains is necessary. Thus, there is a need for an artefact management system such as OSCAR.

Currently OSCAR has been integrated with the Software Configuration Management system, CVS; and there are plans to develop interfaces to various other relevant applications in the software domain, such as PVCS, Rationale Rose, and the Argo UML toolset. In the education domain, interfaces to the contents of various virtual learning environments such as WebCT, FirstClass, and BlackBoard would need to be investigated.

To gain the benefits of widespread reusability, such an artefact management system should be distributed and hosted on an infrastructure such as the Grid to ensure its accessibility and continued development by the international community. The present implementation of OSCAR is being developed as open source, and hosting it over the Grid services toolkit provided by the Globus project [15] would not be problematic.

5. CONCLUSIONS

There are many aspects of collaboration that can be found in any domain where people are working co-operatively supported by computer systems. Here we have considered support developed for artefact management in the context of collaborative software development. By identifying analogous aspects of collaborative learning and education standards being developed, we have shown how this support has the potential to form the basis for more specialised support for eLearning in the context of collaboration over the Grid.

REFERENCES

[1] Foster, I and Kesselman, C. (1999) Computational Grids, chapter 2 in *The Grid: Blueprint for a New Computing Infrastructure*, edited by Ian Foster and Carl Kesselman, Morgan Kaufmann Publishers, Inc., pp. 15-54.

[2] Ellis, C.A., Gibbs, S.J., and Rein, G.L. (1991) Groupware: Some Issues and Experiences, *Communications of the ACM*, Vol. **34**, No. 1.

[3] Gaeta, M and Ritrovato, P. (2002) Generalised Environment for Process Management in Cooperative Software Engineering, *Proceedings of the 26th Annual International Computer Software and Applications Conference*, *COMPSAC02*, IEEE Computer Press, pp. 1049-1053.

[4] Boldyreff, C., Nutter, D., and Rank, S. (2002) Active Artefact Management for Distributed Software Engineering, *Proceedings of the 26th Annual International Computer Software and Applications Conference, COMPSAC02*, IEEE Computer Press, pp. 1081-1086.

[5] Christie, A. M. (1995) *Software Process Automation: the Technology and its Adoption*, Springer.

[6] Aversano, L., Cimitile, A., Gallucci, P., and Villani, M. L. (2002) FlowManager: a workflow management system based on Petrie nets, *Proceedings of the 26th Annual International Computer Software and Applications Conference, COMPSAC02*, IEEE Computer Press, pp. 1054-1059.

[7] Boldyreff, C., Nutter, D., and Rank, S. (2002) Architectural Requirements for an Open Source Components and Artefact Repository system within GENESIS, *Proceedings of the Open Source Software Development Workshop*, University of Newcastle, pp. 176-196.

[8] *IEEE Software Engineering Standards Collection*, IEEE Press, (1994).

[9] Carnegie Mellon University / Software Engineering Institute, (1995) *The Capability Maturity Model: Guidelines for Improving the Software Engineering Process*, Addison-Wesley Publishing Company.

[10] *ISO TR 15504* Software Process Assessment and Improvement (commonly known as SPICE), ISO, (1998) (approved for publication), specifically see Part 2, A Reference Model for Processes and Process Capability.

[11] Kehoe, R., and Jarvis, A.(1996) *ISO 9000-3 a tool for software product and process improvement*, Springer.

[12] Kruchten, P. (2000) *The Rational Unified Process: an Introduction*, Addison-Wesley Publishing Company.

[13] QAA, Quality assurance in UK higher education: a brief guide, The Quality Assurance Agency for Higher Education, available on-line at http://www.qaa.ac.uk/public/heguide/contents.htm

[14] Koper, R. (2001) *Modeling units of study from a pedagogical perspective: the pedagogical meta-model behind EML*, Open University of the Netherlands, First draft, Version 2.
[15] Caeiro, M., Anido, L., Santos, J.M., and Rodríguez, J. (2002) Towards the Standardization of Collaborative Learning Systems, *Proceedings of the Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'02)*, IEEE Computer Press, 2002.
[16] IMS (1999) *IMS Enterprise Information Model*, Version 1.01 - Final Specification, IMS Global Learning Consortium.
[17] Foster, I. and Kesselman, C. (1999) The Globus Toolkit, chapter 11 in *The Grid: Blueprint for a New Computing Infrastructure*, edited by Ian Foster and Carl Kesselman, Morgan Kaufmann Publishers, Inc., pp. 259-278.