Web-Based Support for Managing Large Collections of Software Artefacts

Cornelia Boldyreff, James Brittle, Chris Korhonen, Phyo Kyaw, Janet Lavery, David Nutter, and Stephen Rank
Research Institute for Software Evolution, University of Durham, UK
cornelia.boldyreff@durham.ac.uk

Abstract

There has been a long history of CASE tool development, with an underlying software repository at the heart of most systems. Usually such tools, even the more recently web-based systems, are focused on supporting individual projects within an enterprise or across a number of distributed sites. Little support for maintaining large heterogeneous collections of software artefacts across a number of projects has been developed. Within the GENESIS project, this has been a key consideration in the development of the Open Source Component Artefact Repository (OSCAR). Its most recent extensions are explicitly addressing the provision of cross project global views of large software collections as well as historical views of individual artefacts within a collection. The long-term benefits of such support can only be realised if OSCAR is widely adopted and various steps to facilitate this are described.

Keywords: Software artefact; software repository; web-based support for collaborative development.

1. Background

The systemic representation and organisation of software descriptions, e.g. specifications, designs, interfaces, and implementations, of large distributed application where heterogeneous software components are utilised have been addressed by research within the Practitioner and AMES projects [1,2,3,4]. Without appropriate representations and organisations, large collections of existing software are not amenable to the activities of software reuse and software maintenance, as these activities are likely to be severely hindered by the difficulties of understanding the software applications and their associated components. In both of these projects, static analysis of source code and other development artefacts, where available, and subsequent application of reverse engineering techniques were successfully used to develop a more comprehensive understanding of the software applications under study [5,6]. Later research addressed the maintenance of a web-based component library in the context of component-based software product line development and maintenance [7]. The classic software decompositions, horizontal and vertical [8], proposed by Goguen have influenced all of this research. While they are adequate for static composition, they fail to address the dynamic aspects of composing large distributed software applications from components especially where these include software services that may be dynamically bound at run-time.

Recent research within the CoDEEDS project has made some progress towards the determination of design spaces to support both the static and dynamic system composition as well as the determination of the physical deployment and long-term operation of large distributed system composed from heterogeneous components [9]. Our current prototype implementation of collaborative support for the determination, elaboration, and evolution of design spaces, based on the CoDEEDS framework [10], employs at its base another development of our recent research within the GENESIS project, the Open Source Component Artefact Repository, OSCAR [11, 12, 13, 14].

The GENESIS project's goal is to develop a generalised environment for process management in collaborative software engineering [15]. A key component of this environment is an underlying distributed repository, OSCAR, to hold the artefacts and the meta-data, describing software artefacts, and their contents. OSCAR is currently in the process of being released as a web-based service to support distributed developers of both industrial and open source software development projects [16].

2. Web-based Collaborative Software Development

The web and its associated technologies facilitate communication and cooperation amongst software developers enabling large collaborative software development projects to be undertaken. The open source community provides many examples of such projects. Multinational software projects are also commonplace within industry today. Various solutions are available to address the immediate support of these collaborative development projects throughout the lifecycle of the project. These solutions both open source and commercial vary considerably in the elements of collaborative development and project management they address. SourceForge[1], in the open source domain, provides basic support for managing cooperative development of software artefacts such as handling mailing lists, repository services, and bug tracking. However, it does not support workflow, resource management, or collaborative work by many users on a single artefact (apart from the use of a CVS[2] (Concurrent Versions System) repository to handle configuration management). Microsoft Project Professional[3] supports enterprise project management over single or distributed sites in the commercial domain. It concentrates on the workflow and planning elements of cooperative development but has no specific focus on software engineering projects, unlike Rational's[4] range of products, which support industrial software development across a global enterprise in the commercial domain. There are also general, not software development specific, web-based solutions that have been used support cooperative working of distributed software development teams such as SiteScape[5], which handles a central repository, with forum-like facilities for interaction and BSCW (Basic Support for Cooperative Work) which formed the basis of the SEGWorld development [17]. The GENESIS project has employed SiteScape to manage the deliverables associated with its various work packages and to coordinate document reviewing associated with the project's research and software developments. It is our goal to move to the use of the GENESIS platform with OSCAR for the final stages of the project. All of these current solutions support web-based access to project related data and artefacts under production by the software team.

What is lacking from current solutions is any means of obtaining a global view of project data and software artefacts across a number of projects irrespective of the initial methods and tools employed during the project's lifetime. Here the underlying artefact management system, OSCAR, being developed within the GENESIS project coupled with the CoDEEDS framework offers the basis for delivering such support in the future.

The remainder of this paper is organised as follows: section three describes the overall design of OSCAR and the support for co-operative software development that it currently offers combined with CoDEEDS; extensions to OSCAR to provide historical awareness of artefact development across projects [18] and a global view of a number of distributed artefact repositories are discussed in section four; and finally section five discusses planned deployment and future related research.

3. CoDEEDS, GENESIS, and OSCAR

The CoDEEDS project is concerned with the Collaborative Determination Elaboration and Evolution of Design Spaces [9, 10]. It provides support to design teams enabling them to record their determination of the solution space in the development of large complex distributed systems composed of heterogeneous software components. The result is potentially an N-dimensional design space layered by static and dynamic views of the component sub-systems and models of their deployed instances within the system being designed and deployed in practice. The design environment being developed as part of the CoDEEDS project supports collaborative design throughout the system lifecycle with an agent-based architecture to support design team in their various activities.

Different members of the design team may employ their own preferred design methods and tools when carrying out the detailed design work. The CoDEEDS environment provides a global view of the overall design of the system and the various design decisions that have been made in its composition from a number of potentially heterogeneous components. Figure 1 below indicates the primary areas (use cases) supported by the GENESIS and CoDEEDS system. It shows both the overlapping and discrete primary area.

[1] http://sourceforge.net/
[2] http://www.cvshome.org/
[3] http://www.microsoft.com/
[4] http://www.rational.com/
[5] http://www.sitescape.com/

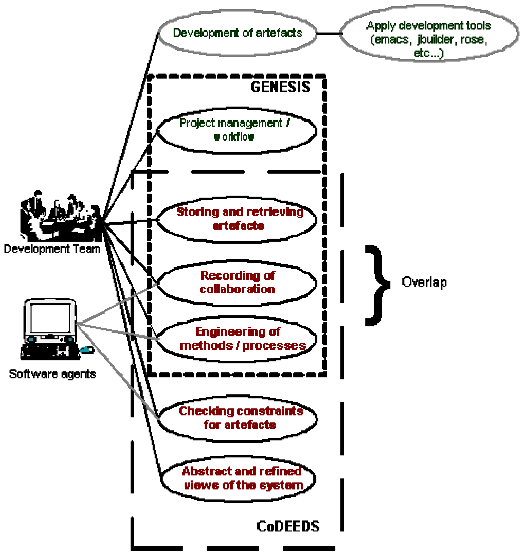
Development of artefacts
Apply development tools (emacs, jbuilder, rose, etc...)
GENESIS
Project management / workflow
Development Team
Storing and retrieving artefacts
Recording of collaboration
Engineering of methods / processes
Overlap
Software agents
Checking constraints for artefacts
Abstract and refined views of the system
CoDEEDS

Figure 1 – GENESIS & CoDEEDS Overlap

The GENESIS project is focused on the development of a Generalised Environment for Process Management in Co-operative Software Engineering. In the context of Figure 1 it addresses the needs for process and work product management. It is employed at both the project management and process workflow level. It complements the design rationale capture of the CoDEEDS system through its support of process engineering and collaborative activity recording. The GENESIS project has developed a low-overhead platform to support distributed software engineering. The system has been designed to be process *aware,* but *non-intrusive*; like CoDEEDS, it does not manadate methods and tools to be employed by the development team. GENESIS is now an open Source project that was seeded by closed initial developments by the project partners.

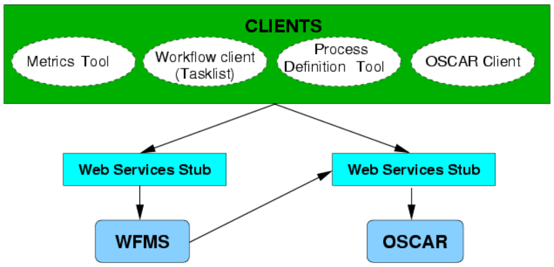

Figure 2 - Overview of GENESIS

GENESIS, outlined in Figure 2, provides a solution for modelling and enacting workflow processes, and for managing both planned and unplanned work products. The process enactment is distributed over multiple physical sites coordinated by a global process at one site. Both local and global processes are managed via the GENESIS workflow management system.

Underlying both the GENESIS platform and the CoDEEDS system is an artefact management system, OSCAR, which acts as a repository for all artefacts resulting from development. OSCAR supports the creation, storage, retrieval, and presentation of artefact data elements and their associated meta-data. “Everything is an artefact” is the view of the repository’s data; this results in a simplified data model throughout OSCAR. By utilising Castor[6] in the implementation of OSCAR, the ability to treat artefacts as objects and documents simultaneously has been achieved allowing for flexible processing and extension of artefacts and their associated types. The actual storage of the artefact content is achieved through plug-in storage mechanisms. An abstraction over software configuration management (SCM) is currently mapped to a CVS plug-in and a plug-in for Perforce[7] is under development. Similarly plug-ins for searching are possible, such as the GENISOM extensions described in the following section. Instrumentation to collect data about the users and system activities provides the basis for awareness extensions also described in the following section.

Currently OSCAR supports the following extensible set of artefact types:

- Software – specifications, designs, code, etc…
- Annotation – any additional information such as email messages
- HumanResource – description of the relevant software engineering personnel
- Project – workflow models and enactment descriptions
- Default – all artefacts are extensions of this

Currently high level artefact operations exist for automatic indexing to support search and retrieval, and for various transformations to allow for flexible presentation of artefacts to users.

4. Historical Awareness and GENISOM

The possibility to extend OSCAR with historical awareness arises along with the cross project historical data that is captured as OSCAR is used to support a number of projects and as data sharing between distributed OSCARs is realised.

[6] http://castor.exolab.org/
[7] http://www.perforce.com/

Historical awareness deals with a collection of heterogeneous artefacts allowing the user to view the complete context of an artefact's creation and history of changes into its present form across a number of projects rather than a contextless view of changes to a single project artefact [18]. Historical awareness is superficially similar to change logs and history views provided by SCM systems but, unlike these systems, provides information that has not been explicitly requested by the user. One way of displaying historical data is via a timeline relating the changes made to an artefact by various users over time. Such a display can be driven by events as they occur providing immediate feedback to developers sharing an artefact across projects or within a single project.

OSCAR's initial large-scale population for demonstration purposes is being derived from the packages of the Debian[8] open source project and will consist of just over 1500 software artefacts. This population with its extracted metadata will be employed in some experimental studies to gauge the effectiveness of using Self-Organising Maps (SOMs) to classify large collections of software artefacts in the GENISOM project [19]. In GENISOM, we have replicated Kohonen's original WebSOM [20] and extended it to the domain of web-based software artefact collections. SOMs are used as a data visualisation technique to support users browsing and searching large collections of data by representing the collection's population as an interactive map. Thereby exploiting computer technology and peoples' abilities to comprehend visual representations. Even though reusable assets are in abundance, a growing problem is the ability to actually locate assets that are relevant for reuse. Organisation of a collection is therefore a necessity and the GENISOM project and other research [21] has come to the conclusion that SOMs are a viable organisational tool that could be used instead of hierarchical or faceted classification. SOMs also provide a virtually automatic organisation process that can save on the costs associated with employing reuse librarians and reduce the amount of time needed to train engineers in the use of the library.

Our preliminary results applying GENISOM to the Debian components show promise and support our belief that SOMs are an ideal solution to organising the incrementally expanding content of the large distributed repositories that we anticipate will result from OSCAR's usage by a growing number of software development projects.

5. Planned Deployment and Future Research

Currently within the framework of the GENESIS project, the consortium's industrial partners are deploying the GENESIS platform including OSCAR in a number of user trials. Members of the GENESIS project team are also actively using it to support their own internal development. A stable version of the GENESIS platform has also been made available on SourceForge [9].

Consideration has been given to advising organisations on the appropriate strategy to employ when adopting the solutions offered by the GENESIS platform. For large organisations with highly distributed cooperating teams the adoption of a new technology is a complicated process that requires an organisation to consider the technology in context of the organisation's business goals [22]. Prior to the adoption of GENESIS a large organisation must determine the answers to two difficult questions. Do the existing software processes require additional or improved technical support supplied by GENESIS? Does the organisation need to improve their software processes and will GENESIS support that improvement effort? It is essential to any organisation that the adoption of any new technology is based on the determined needs of the organisation. In the GENESIS project we advocate the use of the Carnegie Mellon Software Engineering Institute's Capability Maturity Model (SW-CMM) [23] to determine those organisational needs and to support an incremental technology adoption strategy [22]. As GENESIS is a collection of distinct systems that work together to provide effective support for the management of both software product evolution and software processes enactment it is possible to introduce the individual systems incrementally based on the determined needs of the organisation.

To ease adoption of the platform, a stand-alone version of OSCAR has been developed and made available. As well as the tools described earlier to up-load the Debian project software; a simple import tool for Java software and other miscellaneous files has been developed. This has enabled the GENESIS project software to be easily transferred into OSCAR as part of the project's own use of its developments.

[8] http://www.debian.org/

[9] http://sourceforge.net/projects/genesis-ist

As with the local and global work processes, the work products managed by OSCAR will soon be visible in a similarly global name-space composed of multiple local OSCAR repositories. Also in progress for OSCAR is the user-transparent meta-data extraction and indexing functionality.

It is only with the wide spread adoption of OSCAR and the development of much larger collections of OSCAR based software artefacts that advantages, such as being able to obtain global views of such collections held in distributed repositories, will be apparent.

Instrumenting the tools provided by both CoDEEDS and GENESIS will allow studies of software engineering processes and products to be performed. Monitoring the real behaviour of projects managed by the GENESIS workflow engine will allow studies of software development processes, indicating how closely real projects adhere to idealised models.

The architecture of the GENESIS platform currently relies on the relatively tight binding of RMI. This is being transformed to a new architecture based on web services. Once this has been done, the distribution model of the platform will be more flexible. It will no longer be necessary to maintain a strict one-to-one relationship between GENESIS and OSCAR installations; an instance of OSCAR could be shared by more than one GENESIS platform, or a single GENESIS project could use more than one repository.

The industrial partners are currently evaluating the GENESIS project in real projects. The feedback on the prototype platform that is currently being evaluated will provide motivation for future development in terms of functionality, usability, and interaction mechanisms.

GENISOM provides an interesting way to organise of a large collection of artefacts. Work is in progress to evaluate visualisation techniques in terms of their utility to the working software engineering teams.

References

[1] C. Boldyreff, P. Elzer, P. Hall, U. Kaaber, J. Keilmann and J. Witt, "PRACTITIONER: Pragmatic Support for the Reuse of Concepts in Existing Software", *Proceedings of Software Engineering 1990 (SE90)*, Brighton, UK, Cambridge University Press, 1990.

[2] Cornelia Boldyreff, "A Design Framework for Software Concepts in the Domain of Steel Production", *Proceedings of the Third International Conference on Information System Developers Workbench*, Gdansk, 22-24 September 1992.

[3] C. Boldyreff, E.L. Burd, R.M. Hather, R.E. Mortimer, M. Munro, E.J. Younger, "The AMES Approach to Application Understanding: a case study", *Proceedings of the International Conference on Software Maintenance*, IEEE Computer Press, 1995.

[4] C. Boldyreff, E.L. Burd, R.M. Hather, M. Munro, E.J. Younger, "Greater Understanding Through Maintainer Driven Traceability", *Proceedings of the 4th Workshop on Program Comprehension, April 1996*, pp. 100-106, IEEE Computer Press, 1996.

[5] Jian Zhang and Cornelia Boldyreff, "Towards Knowledge-Based Reverse Engineering", *Proceedings of the Fifth Annual Knowledge-Based Software Assistant Conference*, Syracuse, NY, 24-28 September 1990.

[6] M. Joanna Fyson and Cornelia Boldyreff, "Using Application Understanding to support Impact Analysis", *Journal of Software Maintenance: Research and Practice*, 10, pp. 93-110, Wiley, 1998.

[7] O. C. Kwon, C. Boldyreff and M. Munro, "An Integrated Process Model of Software Configuration Management for Reusable Components", *Proceedings of the Ninth International Conference on Software Engineering & Knowledge Engineering (SEKE'97)*, June 18-20, 1997, Madrid, SPAIN.

[8] Joseph A. Goguen, Reusing and Interconnecting Software Components, *IEEE Computer*, pp 16-28, February 1986. Reprinted in *Tutorial: Software Reusability*, edited by Peter Freeman, The Computer Society Press of the IEEE, pp 251-263, 1987.

[9] C. Boldyreff, P. Kyaw, D. Nutter, and S. Rank, "Architectural Framework For a Collaborative Design Environment", *Proceedings of Second ASERC Workshop on Software Architecture*, Banff, Canada, 2003.

[10] C. Boldyreff, P. Kyaw, "A Framework for Developing a Design Evolution Environment", submitted to The 27th Annual International Computer Software and Applications Conference COMPSAC (Hong Kong, 2003).

[11] Cornelia Boldyreff, David Nutter, and Stephen Rank, "Architectural Requirements for an Open Source Component and Artefact Repository system within GENESIS", *Proceedings of the Open Source Software Development Workshop*, Newcastle upon Tyne, U.K. 25-26th February 2002, pp 176-196.

[12] Cornelia Boldyreff, David Nutter, and Stephen Rank, "Open-Source Artefact Management for Distributed Software Engineering", *Proceedings of the 2nd Workshop on Open-Source Software Engineering at The 24th International Conference on Software Engineering in Orlando*, May 2002.

[13] Cornelia Boldyreff, David Nutter, and Stephen Rank, "Active Artefact Management for Distributed Software Engineering", *Proceedings of the Workshop on Cooperative Supports for Distributed Software Engineering Processes*, in the Proceedings of the 26th IEEE Annual International

Computer Software and Application Conference, August 2002.

[14] D. Nutter, C. Boldyreff, S. Rank, "An Artefact Repository to Support Distributed Software Engineering", *Proceedings of 2nd Workshop on Cooperative Supports for Distributed Software Engineering Processes*, CSSE 2003, Benevento, Italy, March 2003.

[15] Matteo Gaeta and Pierluigi Ritrovato, "Generalised Environment for Process Management in Cooperative Software Engineering", *26th Annual International Computer Software and Application Conference Proceedings*, IEEE, 2002, pp. 1049-1053.

[16] Cornelia Boldyreff, Janet Lavery, David Nutter, and Stephen Rank, "Open-Source Development Processes and Tools", *Proceedings of Taking Stock of the Bazaar: 3rd Workshop on Open Source Software Engineering*, Portland, Oregon, May 3 2003.

[17] S. Drummond, C.Boldyreff "SEGWorld: A WWW-based Infrastructure to Support the Development of Shared Software Engineering Artifacts" *Workshop on Web-Based Infrastructures and Coordination Architectures for Collaborative Enterprises*, IEEE 8th Intl. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), IEEE Press, 1999, pp. 120 -125.

[18] David Nutter and Cornelia Boldyreff, "Historical Awareness Support and Its Evaluation in Collaborative Software Engineering" to appear in Proceedings of the Workshop on Evaluation of Collaborative Information Systems and Support for Virtual Enterprises at the 12th IEEE international Workshops on Enabling Technologies For Collaborative Enterprises (WETICE), 2003.

[19] James Brittle, "Self Organizing Maps Applied to Web Content", Final Year Project Report, Department of Computer Science, University of Durham, May 2003.

[20] T. Kohonen, S. Kaski, K. Lagus, J. Salojarvi, J. Honkela, V. Paatero and A. Saarela, "Self Organization of a Massive Document Collection", *IEEE Transactions on Neural Networks*, Vol. 11, No. 3, May 2000, pp. 574-585.

[21] Dieter Merkl, Self-Organizing Maps and Software Reuse, *Computational Intelligence in Software Engineering*, World Scientific, 1998.

[22] J. Lavery, C. Boldyreff, D. Nutter, and S. Rank, "Incremental Adoption Strategy for the GENESIS Platform", GENESIS Project Report, University of Durham, available on-line at

http://www.dur.ac.uk/janet.lavery/documents/AdoptStratFinal.pdf

[23] Mark C. Paulk, Bill Curtis, Mary Beth Chrissis, and Charles V. Weber, "The Capability Maturity Model for Software", *IEEE Software*, Vol. 10, No. 4, July 1993, pp. 18-27.