in: Proc. 8th Int. Conf. on Extending Database Technology (EDBT'2002),
Software Demonstration Session, Prague, March 2002, pp. 745-748

The ORDB-based SFB-501-Reuse-Repository[1]

Wolfgang Mahnke, Norbert Ritter

University of Kaiserslautern
P.O.Box 3049, 67653 Kaiserslautern, Germany
{mahnke|ritter}@informatik.uni-kl.de

Abstract. Comprehensive reuse and systematic evolution of reuse artifacts as proposed by the Quality Improvement Paradigm (QIP) require an integrated management of (potentially reusable) experience data as well as project-related data. This demonstration presents an approach exploiting object-relational database technology to implement a QIP-driven reuse repository. Our SFB-501-Reuse-Repository is designed to support all phases of a reuse process and the accompanying improvement cycle by providing adequate functionality. Its implementation is based on object-relational database technology along with an infrastructure well suited for these purposes.

1 Experience Data Management

Learning from experience gained in past projects is seen as a promising way to improve software quality in upcoming projects. As a result, (anti-)patterns, frameworks, and code fragments are being developed to capture the gained experience of software already developed. But experience is not only represented in the form of (directly) reusable software artifacts. To allow comprehensive reuse [2] a large variety of different reusable elements, e. g., process descriptions or lessons learned [3], are to be managed. Consequently, every kind of (software engineering) experience, independent of its type of documentation, is to be regarded as an *experience element*.

However, the benefits that can be achieved by reusing such experience elements strongly depend on their quality. Thus, they always have to represent the latest state of the art. Hence, checking and continuously improving their quality becomes a crucial issue. The Quality Improvement Paradigm (QIP) [1] suggested by Basili et. al. deals with this problem by integrating systematic evolution and comprehensive reuse of experience elements into an improvement cycle. A QIP cycle consists of several steps, mainly dealing with the planning and executing of an experiment (project), analysing its results, and packaging the gained experience for later reuse. To comprehensively support the overall QIP cycle we conceptually extended the notion of an Experience Base (EB) as introduced in [1]. Our resulting repository structure consists of two logically disjunct sections called Organization-Wide Section (OWS) and Experiment-Specific Section (ESS), where the OWS is an instantiation of Basili's EB and the ESS holds the experiment documentations. To evaluate the usefulness of the conceptual extension, we implemented a web-based prototype of the reuse repository [4]. Thus, we also concentrate on the technological advancements for such QIP-driven reuse repositories. Besides discussing the organizational repository structure and the interface functions required to provide comprehensive support, we demonstrate that the infrastructure provided by new Object-Relational Database Management Systems (ORDBMSs) [6] can effectively be used for realization purposes. As far as we know, our approach is the first one evaluating *object-relational* database technology in the field of reuse repositories.

1. This work has been supported by the Deutsche Forschungsgemeinschaft (DFG) as part of the Sonderforschungsbereich (SFB) 501 "Development of Large Systems with Generic Methods".

2 Our Approach: SFB-501-Reuse-Repository

Representations of *experience elements* (EE) are given in many different data formats. To easily handle the different formats (and to be open for new formats), EE representations should only be saved using a plain data type in the repository without respect to special data formats. ORDBMSs offer a special kind of data type for this purpose, called BLOB (binary, large object). A representation can be composed of several parts. For example, an HTML document (e. g., a framed HTML page) can consist of many files. Therefore, each representation is stored in a set of BLOBs. Also, several alternative representations of the same EE can occur.

For retrieval purposes, each EE needs to be associated with a so-called *characterization vector* (CV) containing describing data, e. g., relevant for (similarity-based) search of potentially reusable design artifacts. For performance reasons, EEs (containing large objects) and corresponding CVs (comparably small amount of data) are stored separately. Whereas all EEs are stored using the same data structure, the CVs are classified into semantically different data structures. CV attributes depend on the section (OWS, ESS) that the corresponding EE belongs to. More precisely, the sections are divided into logical areas, and the areas determine the CV attributes. The object-oriented capabilities of object-relational database technology allowed us to effectively map the mentioned data structures to a database schema.

At its user interface, the SFB-501-Reuse-Repository provides functions specifically supporting the different QIP phases, especially for planning and executing a project as well as analysing project data and packaging results. Several user roles (from *repository manager* to *project team member*) are distinguished. At the beginning of user sessions an authorization component checks security. A user's role, the QIP phase of the experiment (s)he is working for as well as personal access rights determine the functions dynamically provided (to the specific user) at the system interface, the visibility of experience elements (for this specific user) as well as the possible effects of data manipulations (issued by this specific user).

As realization platform, we have exploited the ORDBMS Informix Internet Foundation.2000 [5] including Informix WebBlade as web infrastructure. The database stores EEs, CVs, pre-defined HTML pages as well as extensions (special tables, user-defined functions (UDFs)) needed to dynamically generate HTML pages and answer user requests. In order to answer special user requests, HTML templates including SQL statements can be stored. A user request that has been specified at the browser and passed to the DB Server may address such an HTML template. The UDF *webexplode* (offered by the WebBlade) evaluates the SQL statements contained in the template, incorporates the results into the template and sends it back to the browser. We also exploited the possibility of adapting or even generating the SQL statement(s) to be evaluated in order to create the resulting HTML page dynamically.

3 Focus of Demonstration

To gain experience with the new object-relational technology we have chosen the, as we call it, *extreme extending* (X^2) approach. i. e., almost everything has been implemented by using the extensibility infrastructure of the ORDBMS. Thus, X^2 means that not only the entire application logic runs within the DB Server, but also major parts of the presentation layer (GUI) reside within the DB Server, because HTML pages used for user interaction are dynamically generated within the DBS.

To point out our approach, we focus on similarity-based search primarily used to prepare new experiments by identifying potentially reusable artefacts. First of all, the user specifies a comparison instance by providing comparison values for some CV attributes reflecting aspects which are important w. r. t. the goals of the new experiment. Fig. 1 outlines how the components of the SFB-501-Reuse-Repository collaborate to evaluate

such a query (follow steps a to e) and to deliver a ranked list of EEs, potentially useful for the requesting user.

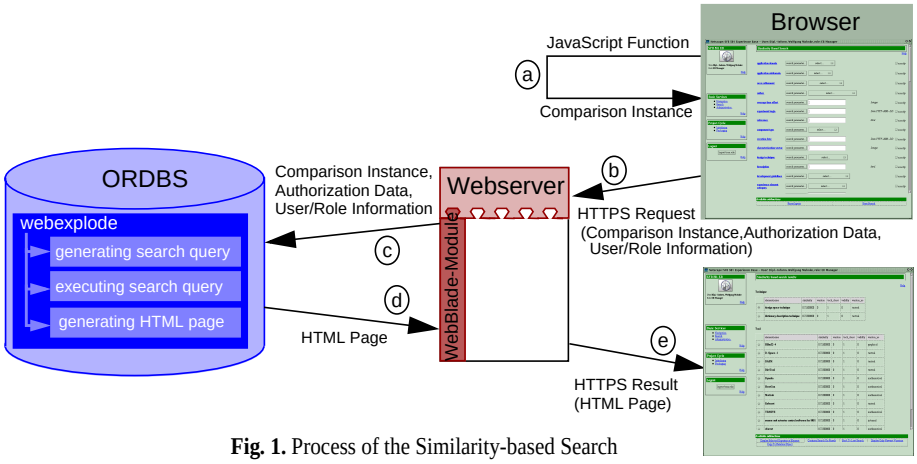

Fig. 1. Process of the Similarity-based Search

This software demonstration of the SFB-501-Reuse-Repository has been designed to: first, point up the capabilities of the system in supporting comprehensive reuse, and, second, to illustrate the (DB-server-)internal processing. For these purposes a simplified software development process is played through and the internal steps of performing similarity search are presented by a visualization component especially implemented for demonstration purposes.
This, on one hand, clarifies that extensibility, which is the key feature of ORDBMSs, can help to effectively implement advanced applications like a reuse repository, but, on the other hand, also poses the question about the reasonable degree of exploiting extensibility, since we feel that counting exclusively on extensibility for the implementation of advanced applications may become counterproductive with a certain threshold of complexity.
Thus, although we want to demonstrate that choosing the X^2 approach has been adequate for implementing our reuse repository, it is not our goal to propagate it unconditionally. Our work also aims at finding rules for a kind of restrained usage of ORDBMS's extensibility features, because at some level of complexity the X^2 approach will definitely lead to a whole bunch of problems, e. g., concerning system performance and robustness as well as ease of development.

Literature

[1] V. R. Basili, G. Caldiera, H. D. Rombach. Experience Factory. In J. J. Marciniak (ed), *Encyclopedia of Software Engineering, Volume 1*, John Wiley & Sons, 1994, 469–476.

[2] V. R. Basili, H. D. Rombach. Support for comprehensive reuse. In *IEE Software Engineering Journal*, 6(5):303–316, September 1991.

[3] A. Birk, C. Tautz. Knowledge Management of Software Engineering Lessons Learned. In *Proc. of the 10th Int. Conference on Software Engineering and Knowledge Engineering (SEKE'98)*, San Francisco, CA, June 1998.

[4] R.L. Feldmann, B. Geppert, W. Mahnke, N. Ritter, F. Rößler. An ORDBMS-based Reuse Repository Supporting the Quality Improvement Paradigm - Exemplified by the SDL-Pattern Approach, in: TOOLS USA 2000, July 2000, pp. 125-136.

[5] Informix Internet Foundation.2000 Documentation, http://www.informix.com/answers/english/iif2000.htm, 2001.

[6] M. Stonebraker, M. Brown. Object-Relational DBMSs - Traking the Next Great Wave. Morgan Kaufman, 1999.