A WEB 2.0 CASE TOOL SUPPORTING PACKAGED SOFTWARE IMPLEMENTATION

Harris Wu and Georges Arnaout
Old Dominion University, Norfolk, VA, U.S.A.

Keywords: Packaged Software Implementation, Web 2.0, Wiki, case-based reasoning.

Abstract: Companies increasingly rely on implementing packaged software instead of developing custom solutions. Packaged software implementation (PSI) is the process to solve business problems by customizing and integrating an off-the-shelf software package. However, there has been a lack of CASE (Computer Aided Software Engineering) tools to support PSI. This paper presents a Web 2.0 based tool that supports case-based reasoning in PSI. The tool helps users explore past design cases, find a similar case, and reuse the design for that case in new problem situations. Our belief is that by utilizing the social power of a large group of users, better designs can be achieved at lower risks and lower costs.

1 INTRODUCTION

One of the trends in information systems development is the increasing use of off-the-shelf software packages. The worldwide ERP (Enterprise Resource Planning) software market is expected to grow from $28.8 billion in 2006 to $47.7 billion by 2011, according to AMR Research. Many other software packages such as scientific and engineering packages also have billion-dollar markets.

Two key issues in packaged software implementations (PSI) are: 1) design exploration: how to arrive at a good design by exploring alternatives; and 2) reuse: how to save the cost and decrease the risks by reusing existing designs. In fact, packaged software implementations frequently practice case-based reasoning: explore the past implementation cases, find a similar case, and reuse the design for that case. As the same software package is being deployed in many implementations, there is a huge potential for designs to be shared, explored and reused across different implementations. However, there has been a lack of CASE (Computer Aided Software Engineering) tools that support design sharing, exploration and reuse.

Our fundamental belief is that by utilizing the collective power of a software community, better designs can be achieved at a lower cost. To explore the potential of the collective power, we are developing and evaluating a Web 2.0 CASE tool that supports collaboration, exploration and reuse in a packaged software implementation community. The tool allows the community to collaboratively create, evolve and classify a repository of cases. The tool allows users to explore the complex design space along multiple perspectives. The tool allows users to collaboratively recommend existing designs to new business requirements.

In the following sections we will first discuss the unique challenges in packaged software implementations, and how case-based reasoning as a design methodology may help address these challenges. Then we discuss how Web 2.0 can be applied towards cased-based reasoning. We present a Web 2.0 CASE tool that is being developed to support packaged software implementations. We report the deployment of the prototype in the community of an open-source software package.

2 CASE-BASED REASONING IN PSI

PSI is different from ground-up software development in many ways:

- There is no design of a complete system. Instead, PSI focuses on pieces of customization and integration.

Wu H. and Arnaout G. (2008).
A WEB 2.0 CASE TOOL SUPPORTING PACKAGED SOFTWARE IMPLEMENTATION.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - ISAS*, pages 343-346
DOI: 10.5220/0001715103430346

- Packaged software customers share similar customization, integration, enhancement, and upgrade needs.
- Instead of the traditional top-down design paradigm, PSI starts from gap analysis to package customization and integration. A significant portion of PSI is matching problems to existing solutions.
- Packaged software implementations often involve external consultants, who bring "the best practice" from implementations for other customers. The ability to leverage consultants' past experience is important to a successful PSI.
- Packaged software implementations usually are joint efforts by several organizations: business users and technical personnel from the customer (the client company), functional and technical consultants from the implementation consultancy, and the software vendor. Inter-organizational collaboration is far more challenging than intra-company collaboration in tradition software development.
- Due to the proprietary nature of packaged software, PSI typically does not use mainstream IDE (Integrated development environment) tools. The availability and applicability of CASE (computer aided software engineering) tools is very limited.
- Due to the limitations and the rigidity of the software package, certain seemingly-simple business requirements may be cost prohibitive or infeasible. As PSI often involves contracts within a fixed price range, lowering the risk is given top consideration among the design choices. Exploring the technical limitations and possible risk areas occurs as early as in pre-sales and contract negotiation.

As seen from above, design exploration and reuse can significantly lower the cost, increase the quality and reduce the risks in packaged software implementations. Exploring and reusing past designs as a methodology is called case-based reasoning (CBR), defined as a problem solving approach that solves a new problem by exploring previous experiences, recalling a similar situation and reusing information from that situation (Maher and Pu 1997). Despite its significant benefits, CBR is largely under-utilized in PSI. The biggest challenge to CBR is to build a rich, up-to-date, explorable and reusable knowledge base of cases. Next section will look at how social power can be utilized to build such a knowledge base.

3 HARNESSING SOCIAL POWER

A new trend on the Web is to utilize the "free brains and labor out there" among Internet users. Harnessing social intelligence, i.e. utilizing individual users' knowledge and efforts, is one of the core competencies of "Web 2.0" (O'Reilly, 2005). We see such phenomenon everywhere now: collaborative spam email filtering, Wikipedia (the largest encyclopaedia) and even CNN.com's attempt to create user contributed news.

The CBR process can potentially harness the social intelligence from various participants in packaged software implementations. Besides documenting and contributing their unique experiences, individuals can help improve each others' designs. Individuals can categorize design cases along multiple dimensions. Their categorizations help draw connections between a new problem situation and existing solutions. Our fundamental belief is that a diverse, large group of people can do better than a small team of experts.

Documenting and collaboratively evolving the cases can be aided by today's Wiki (Web-based collaborative editing) systems. Wiki can allow customers and consultants collaboratively edit the requirement and design documents. An existing solution can be updated for a functional enhancement or compatibility with a new release. While typical project documents are archived after the project completion, the documents in a wiki can evolve after the software implementation goes into production. A past design can be copied, modified and reused in a wiki, perhaps by another team, for another project. The ease of contribution and the informality lead to success as in Wikipedia.

Searching for cases can be aided by social tagging. Emerging social tagging systems such as del.icio.us and flickr.com allow users to assign free-formed keywords (tags) to any documents, and share these tags. Any users can browse these tags, or search for documents tagged with given keywords. While traditional repositories require objects to be labeled with metadata at the time of creation or by privileged administrators, social tagging allows users label objects not created or owned by them, and share these labels. In other words, the metadata is created by data consumers versus data producers. The connections such drawn between objects and concepts (tags) are often beyond what is conceived at the time of creation or by any central indexers.

Exploration of a complex design space can be made easier if the past design cases are categorized along multiple perspectives, i.e. classified along

multiple facets. For example, design documents can be categorized by software product, release, geography, business process, technology, integration interface, etc. Although good for casual discovery and serendipitous reuse, tags do not support systematic exploration as hierarchical structures do. Some recent research allows users to collaboratively classify documents into a predefined faceted schema (Quintarelli et al., 2006), or automatically build a hierarchical schema from tags (Schmitz and Patrick, 2006). Furthermore, it is possible to allow users to edit and evolve the faceted schema in a wiki fashion (Wu et al., 2007).

Members of a PSI community can recommend solutions to each other's new business problems, or point out similar problem situations. Automatic recommendation is also possible, just like the book recommendations by amazon.com. For collaborative recommendation to succeed, users need to formulate their problems well. Categorizing both the problems and solutions will help. In addition, it is important to capture feedback on user- or system-provided recommendations.

4 THE WEB 2.0 CASE TOOL

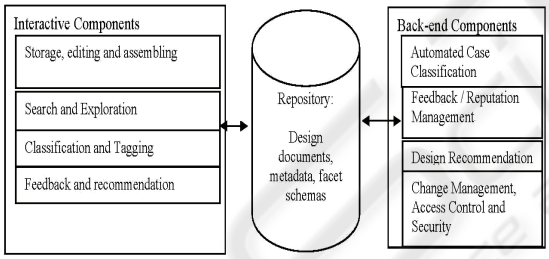

Figure 1: Major Components of the Tool.

Figure 1 shows the major components of the Web 2.0 CASE (Computer-Aided Software Engineering) tool that we are developing. While traditional software engineering relies on general knowledge of a problem domain, case-based reasoning utilizes the specific knowledge of concrete cases. The center of the CASE tool is a repository of cases along with the metadata on these cases. The interactive components are Web-based and only require users to have a Web browser. The tool is implemented on the LAMP platform: Linux, Apache, MySQL and PHP.

To reduce the efforts "reinventing the wheel", particularly the user interface development efforts, the tool integrates several popular open source components (Figure 2) in addition to custom coding.

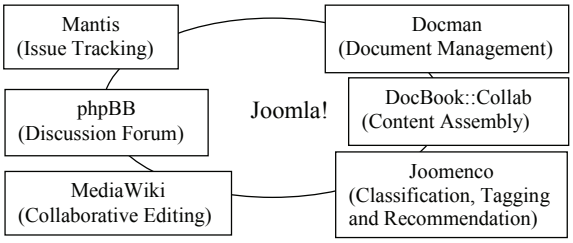

Figure 2: Integration of Open Source Components.

We choose Joomla, a popular open-source content management system (CMS), as the container for our CASE tool. We want to use an open source CMS as it is appropriate for a knowledge repository and content management. Also, popular open source CMS tools have thousands of developers and large numbers of extensions with rich features. For example, Joomla has a Community Builder extension (not included in Figure 2) that supports user profile and reputation management, which saves our own efforts in developing similar features. Other popular CMS we have considered include Drupal, Plone and Zope, which are almost equally good. We choose Joomla as its community seems to be most vibrant. Joomla won the best CMS award by PacktPub.com in 2007.

Mantis, MediaWiki and phpBB are the best-of-breed open source tools in their respective arenas: issue tracking, wiki, and discussion forum. Issue tracking/resolution tools are used in about every packaged software implementation, for testing, go-live preparation and post-live maintenance. Wikis and discussion forums are increasingly used in PSI, particularly the implementations of open source software. By integrating Joomla with these tools, the knowledge and experience embedded in the discussions and issue resolutions can be utilized as "cases" to support problem solving in the future. It is worth noting that there already exist Joomla extensions that integrate Joomla with Mantis, MediaWiki and phpBB. Similarly, the existing Docman extension provides a number of file management features such as uploads/downloads, search, category management and access control. Files, Joomla content items, discussion threads, issues and resolutions are all considered as knowledge cases in the repository.

The Docbook:Collab component allows collaborative editing, document assembly and also import/export of Joomla content to XML and various other formats. Import/export is important to integrating the front-end with back-end techniques that further manipulate the user-contributed content.

We are adding more features to make it usable for modular document collaboration and case encapsulation. We are also fixing this component for browsers other than FireFox/Mozilla as it was originally limited to.

We created the Joomenco component to allow users to explore the case repository using a faceted classification (Hearst, 2006). Furthermore, Joomenco allows users to collaboratively build the faceted schema and classify documents into this schema, in addition to tagging documents with free-form keywords. We also allow users to recommend a solution to a problem, essentially by tagging the document containing the problem situation with a link to the relevant solution.

In addition to development and integration of the aforementioned user interface components, we are developing server-side features such as automatic classification and recommendation. At this stage we are experimenting with different underlying algorithms.

5 PROTOTYPE DEPLOYMENT AND EVALUATION

A preliminary prototype of the tool has been deployed at the community of PostBooks, a full-fledged open-source ERP software package. In addition to the major components discussed above, a bridge is added between Joomla and Sourceforge so that issues and discussions can be transferred seamlessly between them.

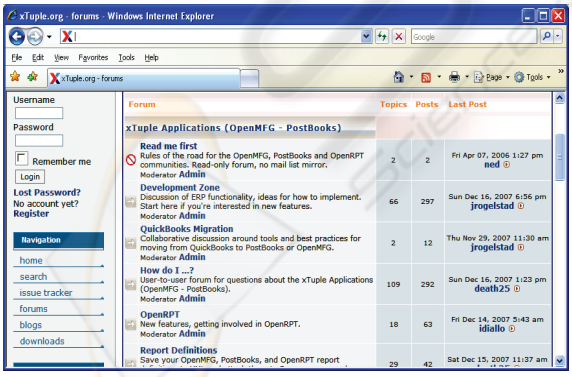

Figure 3: Joomla-based Interface of the Prototype.

After the deployment of the prototype, the user participation has seen a significant increase. PostBooks ranked as #10 on Sourceforge.net in December 2007, up from around #60 prior to the deployment of the tool. User participation increased, likely because now they see the benefit of utilizing the case repository to find solutions, and therefore they are more motivated to contributing problem descriptions and solutions. As one developer noted, "we didn't tell people to do this, they're just doing it." We will survey users on whether and how much the system has saved their efforts or improved quality of their implementations.

As formative evaluation, the deployment has suggested priorities in developing and improving the tool. For example, we are repackaging the Joomenco component (to remove custom code dependencies), and plan to release the component soon to the open source community. We are adding feedback mechanisms to various components of the tool, to capture what users have considered to be useful or not useful. We are working on algorithms that provide real-time recommendations to user-entered problem situations. We are also improving the access control of the tool to address confidentiality, privacy, and security issues.

ACKNOWLEDGEMENTS

This project is supported in part by the United States National Science Foundation, Award No. 0725277.

REFERENCES

Hearst, M.A., Clustering versus Faceted Categories for Information Exploration. Communications of the ACM, 2006(49(4)).

Maher, M. L and Pu, P. Issues and Applications of Case-Based Reasoning in Design. Lawrence Erlbaum Associates, Mahwah, NJ, 1997

O'Reilly, What is Web 2.0, September 30, 2005.

Quintarelli, E., L. Rosati, and Resmini, A. Facetag: Integrating Bottom-up and Top-down Classification in a Social Tagging System. 2006.

Schmitz and Patrick, Inducing Ontology from Flickr Tags. Workshop in Collaborative Web Tagging, 2006.

Wu, H., Zubair, M. and Maly, K. "Collaborative Classification of Growing Collections with Evolving Facets," ACM Hypertext and Hypermedia Conf. 2007.