An Infrastructure for Indexing and Organizing Best Practices

Liming Zhu[1,2], Mark Staples[1,2], Ian Gorton[3]
[1]*NICTA, Australian Technology Park, Eveleigh NSW, Australia.*
[2]*School of Computer Science and Engineering, University of New South Wales, Australia*
{Liming.Zhu, Mark.Staples}@nicta.com.au
[3]*Pacific Northwest National Laboratory*
ian.gorton@pnl.gov

Abstract

Industry best practices are widely held but not necessarily empirically verified software engineering beliefs. Best practices can be documented in distributed web-based public repositories as pattern catalogues or practice libraries. There is a need to systematically index and organize these practices to enable their better practical use and scientific evaluation. In this paper, we propose a semi-automatic approach to index and organise best practices. A central repository acts as an information overlay on top of other pre-existing resources to facilitate organization, navigation, annotation and meta-analysis while maintaining synchronization with those resources. An initial population of the central repository is automated using Yahoo! contextual search services. The collected data is organized using semantic web technologies so that the data can be more easily shared and used for innovative analyses. A prototype has demonstrated the capability of the approach.

1. Introduction

Industry best practices are widely held but not necessarily empirically verified software engineering beliefs. Best practices can be documented in distributed web-based pattern catalogues [2, 11] and practice libraries [10]. The practices in these repositories are often generic, informal and have only anecdotal justification. However, they are relevant and widely consulted by industry practitioners in their day to day practice. These best practice repositories are often built autonomously and scattered around the Internet in free text formats with little structure. They may be maintained by an organization, an individual, or by the general public. For practitioners, it is often difficult to systematically explore repositories for a particular domain, and more difficult to make informed choices among the practices. For researchers, it is currently infeasible to conduct rigorous research on them as a collective whole.

Providing an infrastructure for organizing these practices can contribute to an understanding of how they are created and used, and the role of evidence in justifying them. However, there are a number of challenges in collecting these many practices into a large-scale central repository:

1) The information is constantly evolving. Any harvesting through copying either destroys the up-to-date link with the original resources or involves costly synchronization.

2) Best practices are usually described in a way that can make them relevant to multiple organisations, but can still contain technology-specific or domain-specific information. There is a need to store information about best practices, either to categories or generalise to more abstract practices, e.g. for meta-analysis, or to specialize to more specific practices, e.g. for instances within individual companies.

3) Exhaustive manual harvesting can be prohibitively costly considering the scale of the information. There is a need to index and organize them in a semi-automated way. Initial automation can help to address the "chicken and egg" problem, bootstrapping the central repository to allow it to provide some immediate benefits to encourage the investment of further human effort.

In this paper, we propose an approach for automatically indexing and organizing best practices. The central repository acts as an information overlay on top of existing repositories. This information overlay provides navigation, organization and retrieval capabilities and can be further optimized manually for different research and practice purposes. It is also intended to help researchers understand the problems practitioners face and the level of evidence they rely upon when choosing best practices. A prototype has been implemented using technologies involving the semantic web, collaborative tagging, and AJAX (Asynchronous JavaScript and XML).

This work is a first step in rigorously combining and analyzing these web-based, distributed and autonomous best practice repositories. We consider the work important because:

1) These practice repositories are widely used in industry but have never been analyzed collectively for particular technology domains. Our approach provides an infrastructure for further analysis.

2) The work proposes a new way of building a central repository from bottom-up by leveraging existing resources. It complements the traditional top-down approach of planning repositories [3].
3) The initial population of the repository is constructed in an automated way. This provides immediate benefits. We believe the cost of populating a repository to a useful degree is a major barrier in building repositories. Our approach addresses this barrier.

In the remainder of this paper, we first discuss the background knowledge and implementation technologies. Then we introduce our approach in section 3 and describe a prototype implementation. We discuss the potential benefits and conclude in section 4.

2. Background

2.1. Experience Repositories

A repository for best practices is a form of experience repositories. There are different types of experience repositories.

Repositories within companies tend to be specific with rich set of context information relevant to a specific organization, and are usually private. Analyses of such repositories have produced valuable insights into how private experience repositories work within an organization [5, 7, 12]. However, the scale of such analyses can be limited due to commercial sensitivity.

Academic-oriented repositories are usually small-scale and rely on a group of core participants to grow and maintain them. They are very useful in conducting rigorous empirical studies. But such repositories and associated research results are less exposed and have less perceived value to industry [9].

Experience repositories on the Internet, such as best practice repositories, are widely used by practitioners but are under-researched partly due to their informal, unstructured nature and large scale. However, the need for understanding informal and unstructured repositories has been well-recognized [5, 7]. Our approach addresses this problem.

Another problem within current experience repository research is that most research focuses on technology support for building repositories rather than how experience sharing actually occurs, partly due to the lack of comparative studies among different approaches to experience sharing [6]. An infrastructure for comparative studies is needed.

We could continue to build better software engineering repositories for increased industry exposure and relevance. However, the approach demonstrated in this paper looks at the problem from a new angle, by organizing existing industry repositories, especially the ones in the public domain, with a view to their further empirical analysis.

2.2. Semantic Web, Collaborative Tagging and Facets

The semantic web is a global mesh of information linked in such a way as to be easily processed by machines. It provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. It models data in a structured way, allowing metadata to be attached to information. Due to these characteristics, we have used semantic web technologies to organize the central index repository. A number of such technologies exist, such as OWL (Ontology Web Language), RDF (Resource Description Framework) and topic maps. We chose topic maps [4] as the structuring technology for our repository, partly because its heritage in information indexing is more closely aligned with our approach.

Collaborative tagging has recently become very popular in the web community as a lightweight way to organize knowledge. Tagging is a way of attaching simple metadata to existing resources identified by a URI. Individuals can define and use their own tags to suit their own needs. The resulting collective set of tags defined by multiple individuals will capture different interpretations for the same thing, and common agreements are indicated by tag frequency. In our approach, we choose to use tags as a way of indexing and organizing best practices. Individual pieces of best practices can be tagged in different ways for different research purposes. We have also provided an initial set of automatically generated tags to reduce the cost of establishing an initial repository.

Tags are usually organized in a flat way, though some services such as tagcloud.com, provide ways to group them to form “tag clouds”. Such organizing capabilities are very limited. We introduce the concept of “facet” in our approach. A facet is a group of tags, but itself is a tag as well. This recursive nature provides a hierarchical way of organizing metadata in a very flexible way. But using tags and facets, we can build complicated structures among pieces of best practices in an evolutionary fashion without hard coding any structures up front.

2.2. Implementation Technologies

2.2.1. TM4J TM4J (Topic Maps for Java) [13] is an open source topic map engine for processing XML-based topic map files. We use it along with TM4Web as the core components behind our prototype.

2.2.2. AJAX While doing tagging in a web-based environment, one significant challenge is on how to efficiently handle potentially thousands of tags/facets on

the UI level. We want to effectively prompt users with existing tags to avoid slightly different tags (e.g. plurals) being used when they mean the same thing. We use the latest AJAX Web 2.0 technology to provide a real-time interactive tagging experience without refreshing web pages.

2.2.3. Y!Q (Yahoo! Contextual Search) Automatically populating a repository with "good enough" tags is important for reducing the initial cost of creating an index repository. Y!Q by Yahoo! provides the ability to perform a "contextual search". The Y!Q web service, when given a URL, will return a list of most relevant metadata. The returned data is not simply a list of keywords, but metadata obtained by leveraging contextual information during searches conducted by millions of users. The list of metadata has already implicitly benefited from collective human user expertise. After inspecting the quality of the metadata returned, we feel that it is good enough for initial fully automated tagging, especially considering that almost no cost is required.

3. An Infrastructure for Indexing and Organizing Best Practices

3.1 Best practice sources

To demonstrate how such a central repository is built, we have selected a specific domain – software architecture and design pattern catalogues and best practice libraries. There are hundreds of such repositories over the Internet, maintained by communities, organizations, companies and the general public. We briefly describe two such repositories:

3.1.1. patternshare.org PatternShare.org is a wiki that aims to document design best practices with associated examples and case studies. Each pattern exists on a single wiki page. These patterns are simply organized into a two dimensional navigation table, and are provided with text-based search functionality.

3.1.2. Corej2eepatterns.com Core J2EE Patterns is website that illustrates best practices for the J2EE technology. It is maintained by the author of the "Core J2EE Patterns" book [1]. Although targeting Java, many of the practices are nonetheless technology independent, and can be applied to many types of enterprise applications. It has many practices in common with patternshare.org, and other intricate relationships exist between the repositories.

3.2. Approach

There are a number of steps involved in our approach.

Step 1: Pre-configuration In order to be more suitable to the targeted domain, the repository can be configured with a simple information structure before being used. In this case, we configure the repository to have three main types of knowledge: Patterns, Non-Functional Requirements (NFR) and Tactics. Design tactics are atomic techniques usually employed in a design pattern. A design pattern has more contextual information and often uses more than one tactic. Essentially, a pattern helps or hinders certain non-functional requirements through tactics. The rationale embodied in best practice is captured in relationships among these three types of knowledge. These rationale are essentially the perceived evidence that why a best practice works. In this case, the repository is pre-configured to capture such evidence. The details of such relationships are not the focus of this paper. Further details are provided elsewhere [14]. For other application areas, the repository can be configured in other ways before initial harvesting, to better support domain specific structures. However, we would suggest that most of the indexing and organization be organised through tags and facets rather than using fixed structures, to enable more flexibility for future repository evolution. Figure 1 shows the relationships among the initial structure, tags, and facets for our example domain.

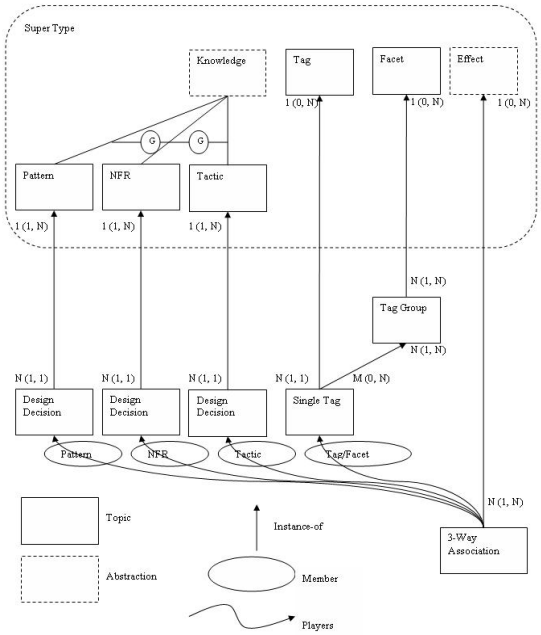

Figure 1. Repository Structure

Step 2: Populating repository The central repository can be populated by providing a collection of URLs or URL patterns for a type of knowledge in a particular best practice repository. The system will submit these URLs to Y!Q contextual search service and retrieve a list of relevant metadata. Users can choose to inspect this metadata before committing the results to the repository.

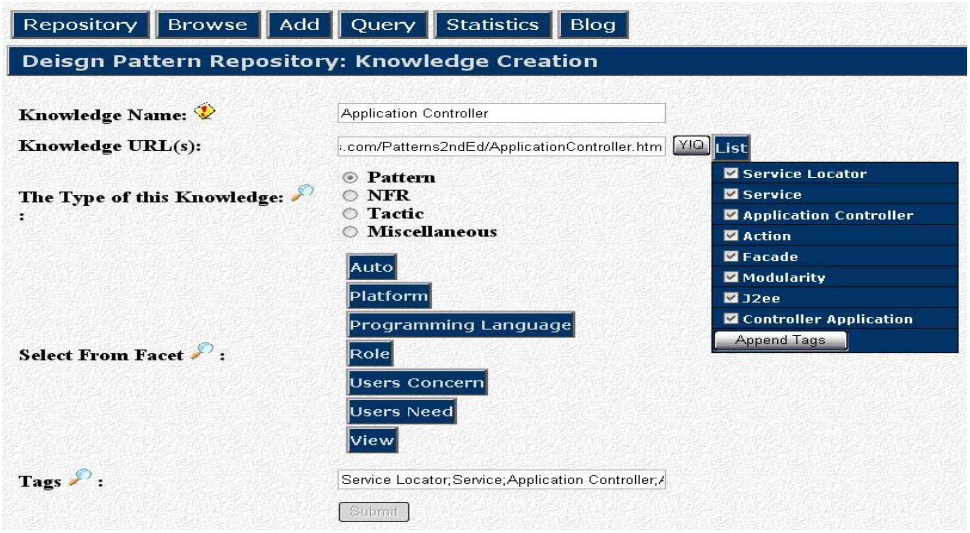

Figure 2 Populate repository

Figure 2 shows an interface supporting this automation and manual metadata inspection. When not using manual inspection, the system can index and organize a large collection of source repositories in a matter of hours.

In the simplest scheme, the title of a page will become the name of the new piece of knowledge. Metadata returned from Y!Q will be the initial set of tags. The URL acts as the unique identifier of this piece of knowledge. When hundreds of such pieces of knowledge are collected, shared common tags will automatically link them together to form a richer structure than the original form. An ontology emerges from such structures.

Step 3: Optimize repository After the initial automated population and metadata attachment, the repository can provide immediate benefits by allowing navigation and search using tags, as shown in Figure 3.

New tags can be added. Hierarchical tags can be organized using facets. New 2-way or multiple-way relationships can be built among knowledge types. Tags can also be attached to these relationships, as shown in Figure 4. Other types of notes (e.g. for meta-analysis purposes) can be also attached. This tagging and annotation can be either done collaboratively, or within an individual's information overlay space for later aggregation.

Figure 3 Populated Repository

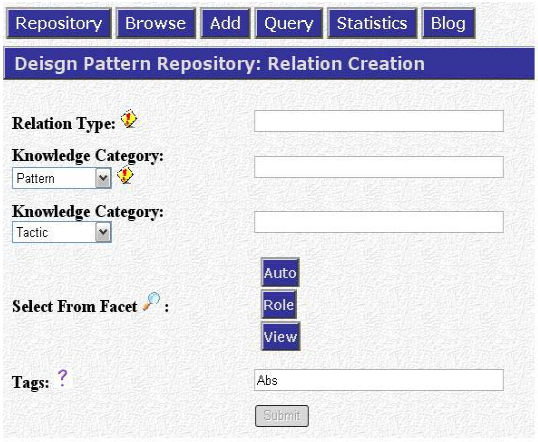

Figure 4 Build Relationships

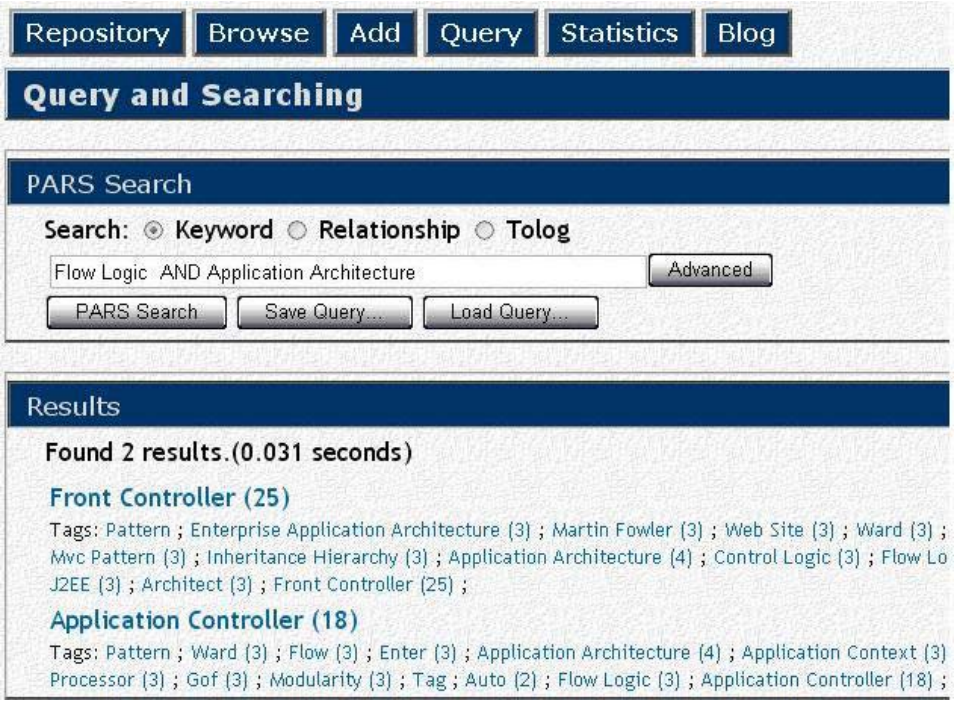

Figure 5 Search the repository

Step 4: Use the repository Depending on the purpose, the repository can be used in different ways. We consider the main contribution of the work as providing an infrastructure for automatically indexing and organizing best practices in a central repository on a large-scale using information overlay.

As an example shown in Figure 5, a search can be performed using keywords, relationships or Tolog [8](a reasoning language) since the repository is essentially organized in an ontology model. In this example, the user wants to retrieve patterns or tactics related to "flow logic" for application architecture. Two results are returned with further metadata descriptions available.

4. Conclusion

The main contribution of our work is to propose a new way of building a central repository from bottom-up to: 1) facilitate further rigorous research for best practice repositories, and 2) understand the problems practitioners face while selecting best practices. We demonstrated the idea by using a prototype in the design best practice domain.

The prototype offers a semantic-web way to organize the information created, which could be used for a variety of purposes. We speculate that there are a number of ways in which such a repository infrastructure could be further used:

1) Conducting large-scale meta-analysis of best practices.
2) Annotating existing repositories more effectively for research purposes.
3) Providing new methods of navigation or search for practitioners.
4) Providing reasoning capabilities in terms of selecting best practices and understanding trade-offs.
5) Conducting research collaboratively.
6) Contributing to a central repository while allowing distributed repositories to be continuously built and evolved autonomously.

Our prototype provides an initial demonstration of the feasibility of the approach, the effectiveness of the automated metadata population and the flexibility for further usage through collaborative tags and facets. We are currently trying to further evaluate the effectiveness of the approach in number of ways:

1) We are examining the quality of the metadata populated automatically or manually.
2) We are trialing its effectiveness in documenting design evaluation evidences in the defense domain.

5. Acknowledgements

NICTA is funded by the Australian Government's Department of Communications, Information

Technology, and the Arts and the Australian Research Council through Backing Australia's Ability and the ICT Research Centre of Excellence programs.

6. References

[1] D. Alur, J. Crupi, and D. Malks, *Core J2EE patterns : best practices and design strategies*, 2nd ed. Upper Saddle River, NJ: Prentice Hall PTR, 2003.
[2] D. Alur, D. Malks, and J. Crupi, "http://www.corej2eepatterns.com," 2003.
[3] V. Basili, M. Lindvall, and P. Costa, "Implementing the Experience Factory Concepts as a Set of Experience Bases," in Proceedings of the 13th International Conference on Software Engineering & Knowledge Engineering, 2001.
[4] M. Biezunski, M. Bryan, and S. R. Newcomb, "ISO/IEC 13250:2000 Topic Maps: Information Technology -- Document Description and Markup Languages," 1999.
[5] T. Chau and F. Maur, "A case study of wiki-based experience repository at a medium-sized software company," in Proceedings of the 3rd International Conference on Knowledge Capture, 2005.
[6] T. Dingsøyr, "An Evaluation of Research on Experience Factory," in Proceedings of the 2nd International Workshop on Learning Software Organization, 2000.
[7] T. Dingsøyr and E. Røyrvik, "An empirical study of an informal knowledge repository in a medium-sized software consulting company," in Proceedings of the 25th International Conference on Software Engineering, 2003.
[8] L. M. Garshol, "Tolog- a Topic Maps Query Language," in Proceedings of the Topic Maps Research and Applications (LNCS 3873), 2006.
[9] B. Kitchenham, T. Dybå, and M. Jørgensen, "Evidence-based Software Engineering," in Proceedings of the 26th International Conference on Software Engineering (ICSE'04), 2004.
[10] Microsoft, "http://msdn.microsoft.com/practices/."
[11] Patternshare.org, "http://www.patternshare.org."
[12] L. Scott and R. Jeffery, "The anatomy of an experience repository," in Proceedings of the International Symposium on Empirical Software Engineering (ISESE'03), 2003.
[13] TM4J.org, "http://www.tm4j.org/."
[14] L. Zhu, M. Ali Babar, and R. Jeffery, "Mining Patterns to Support Software Architecture Evaluation," in Proceedings of the 4th Working IEEE /IFIP Conference on Software Architecture, 2004.