

The Reengineering Wiki

Arie van Deursen

CWI, P.O. Box 94079
1090 GB Amsterdam, The Netherlands
<http://www.cwi.nl/~arie/>

Eelco Visser

Institute of Information and Computing Sciences, Utrecht University
Utrecht, The Netherlands
<http://www.cs.uu.nl/~visser/>

Abstract

The reverse and reengineering research communities have a strong tradition of collecting, organizing, and unifying research results. Typical examples include an explicit taxonomy, dedicated web sites, an annotated bibliography, as well as efforts in exchange formats and tool evaluation. In this paper we describe and evaluate the use of a web authoring system to integrate such efforts. To that end, we propose the “Reengineering Wiki”, which uses Wiki technology to enable web site visitors themselves to maintain and organize pages devoted to their topics of interest. This paper covers web authoring criteria, an introduction to wiki technology, typical wiki usage, and an evaluation of wiki-based systems. Moreover, the paper discusses the organization and contents of the Reengineering Wiki, and concludes with an invitation to participate in the Reengineering Wiki project.

1. Introduction

The software reengineering community has a tradition of recognizing the importance of collecting, organizing, and unifying research results. Some typical examples are:

- the reverse engineering taxonomy [2];
- the reengineering research infrastructure [7];
- several repository sites collecting links in the areas of, e.g., program understanding¹ and decompilation²
- an annotated bibliography [1] integrated with an on line³ bibliography ordered according to the taxonomy of [2].
- the identification of software renovation research directions [3];
- the design of a common exchange format for reengineering tools [5];

¹www.ai.univ-paris8.fr/UPU/repository.html,
www.cs.ualberta.ca/~kenw/toolsdir/

²<http://www.itee.uq.edu.au/~csmweb/decompilation/>

³www.informatik.uni-stuttgart.de/ifi/ps/reengineering/index.html

- the experimental validation of the effectiveness of a range of different reengineering tools [8].

A largely unexplored opportunity for putting together reengineering results from different research groups is the use of web-based collaboration tools. In this paper we will study how such tools could help to collect announcements of important reengineering events, to get a grasp on new developments and trends, to understand relationships between specific reengineering research topics, to keep up to date with interesting papers in particular areas, to ask (clarification) questions, and to pose critical remarks and opinions concerning reengineering developments. A site containing such information could strengthen collaboration in the reengineering community, and could enhance the visibility of the reengineering research results.

The construction and maintenance of such a site, however, can be a major effort. In order to be successful, several criteria must be met:

- The site must be self-maintainable, i.e., there should be no need for a central maintainer responsible for the majority of the work.
- It must be *structured*, allowing for instance, the creation of (multiple) taxonomies;
- It must be *evolvable*, permitting changes to existing pages as well as to the underlying navigation structure;
- It must be *open* to anyone interested in reengineering;
- It must be *easy to use*, both for reading and for contributing.

In this paper we will look at the possibilities offered by so-called *wiki*-based web collaboration tools in order to meet these desiderata. Several sites are successfully running *wiki* servers. The first and still most influential *wiki* site is the one set up by Ward Cunningham,⁴ a site that has done a lot to develop the *design patterns* and *extreme programming* communities.

⁴See www.c2.com/cgi/wiki?WelcomeVisitors

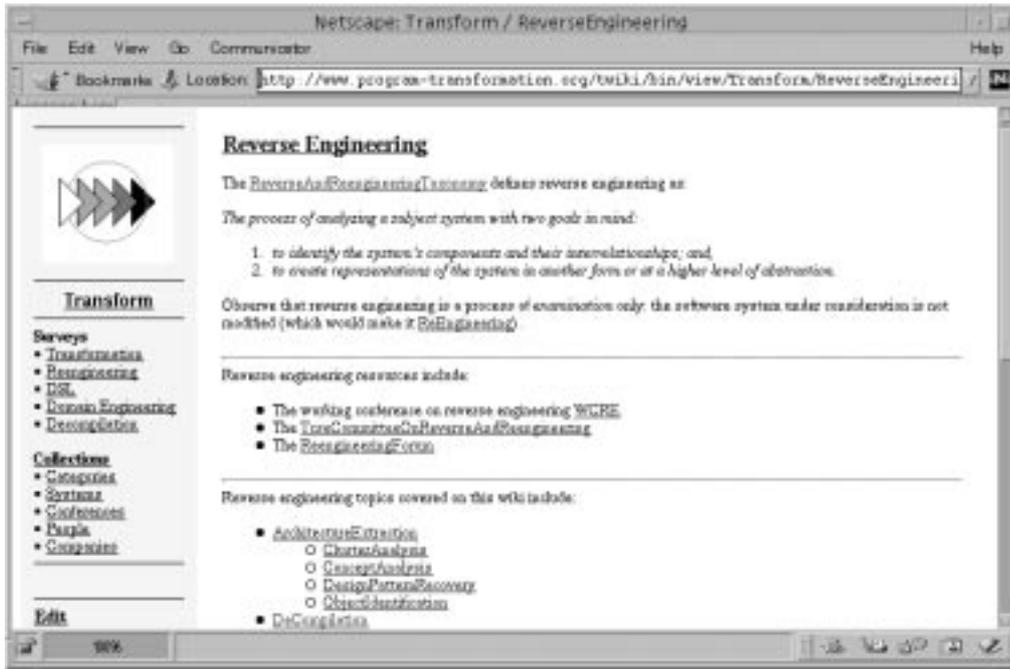


Figure 1. Example Reengineering Wiki Session

In Section 2, we explore Wiki technology, the way in which Wiki systems are used, and compare them to alternative technologies. In Section 3, we discuss our experience in setting up the *Reengineering Wiki*,⁵ analyze the sort of contents that could be included, and explore how this forum could strengthen the reengineering research community.

2. Introducing Wiki

A wiki web is a site that is maintained by the site users: anyone visiting the site can also edit the pages. Wiki wiki is Hawaiian for “quick”, indicating the intent that editing and maintaining the site should be as easy as possible. At the time of writing there are numerous wiki engines available and hundreds of wiki forums discussing many different topics.⁶

2.1. Wiki Technology

Wiki is a simple web authoring tool. A wiki page is edited using a plain text format, which can be automatically translated to formatted HTML. Several intuitive conventions for formatting are used such as: a blank line starts a new paragraph, *_word_* yields an *emphasized* word and **word** a **bold** word, and lines starting with a tab and an asterisk are considered as itemized lists. As an example, Figure 1 displays the HTML corresponding to the wiki text of Figure 2.

Each page has a *wiki name*. Wiki names are constructed by concatenating capitalized words, as, for example *ReverseEngineering*. Wiki names occurring in a page are automatically translated into hyperlinks to the corresponding wiki page. If the page does not yet exist, a hyperlinked question mark is inserted after the wiki name, which can be used to create the page.

Each wiki page contains an edit button. Using this, an HTML text editing area pops up, showing the plain text representation of the page. Upon saving, the modified page is stored. In most wiki systems, there are no access restrictions to the use of the edit button: users browsing through a wiki system are encouraged to add or modify contents whenever they feel the desire to do so.

Navigating through a wiki system is simplified by the fact that links are bi-directional: by clicking on the title of a page all “back links” to that page are displayed. Furthermore, most wiki systems come with regular expression based search capabilities through the contents or names of wiki pages. Changes to as well as creations of pages are reported in a special *RecentChanges* page, providing an overview of what has happened to the site. Some types of servers support notification of (digests of) changes by email.

A wiki *server* maintains the database of wiki pages. The server typically puts pages under revision control, so that in case of mistaken (or malicious) editing, old pages can be restored.

⁵See www.program-transformation.org/re/

⁶See www.c2.com/cgi/bin/wiki?WikiForums.

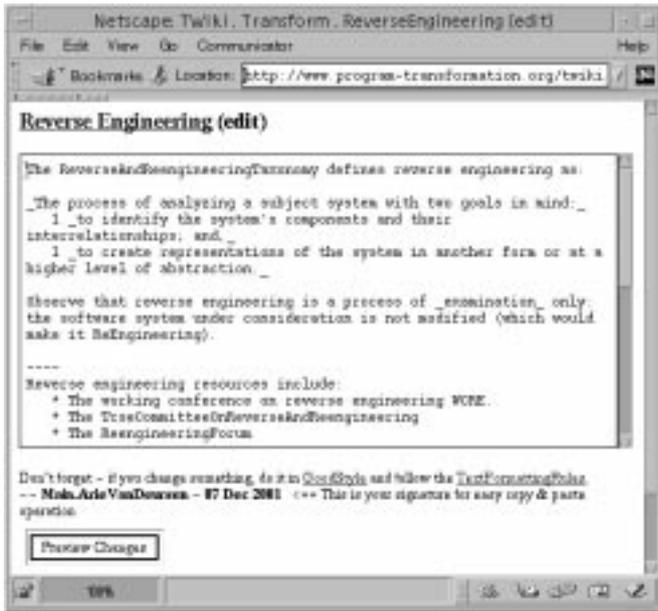


Figure 2. Example Reengineering Wiki Editing Session

2.2. Wiki Usage Patterns

Wiki systems can be used in many different ways. In *document* mode, a wiki page is used to describe the page's topic, for instance by giving a definition, description, or an "opening statement." Wiki users can add comments, or react to the statement.

Comments are often written in *thread* mode. This consists of a sequence of reactions, which are usually separated by a vertical line and signed by the author. Thread mode makes it easy to follow the flow of a discussion, whereas document mode makes it easy to organize information.

In both modes, references to related topics are easily made by just concatenating the appropriate words in the sentence in order to turn them into a link.

To simplify browsing a wiki, pages are typically grouped into *categories*. For this, the standard wiki mechanisms are used: a category is itself a page, a page belongs to a category if the category name is mentioned, and requesting all back links to the category name yields all pages in the category.

Wiki pages evolve, and after a period of significant editing, or when a discussion stabilizes, some form of refactoring may be appropriate in order to render the information in the most accessible way. This may involve creating additional pages devoted to subtopics, renaming pages, reorganizing pages, introducing a new category, and so on. Such refactorings, are typically done by the main contributors of a collection of pages, or by a specially assigned *Wiki Master*.

2.3. Wiki Pros and Cons

A short comparison with other approaches will help to understand the pros and cons of wiki systems. A *mailing list* is suitable for announcements or events of which a subscriber wants to be informed immediately. Like a *news group* it is difficult to structure information, supporting mostly *thread mode* for discussions.

Hand-writing a web site offers full flexibility, but has the key disadvantage of relying on a single maintainer. This problem is addressed by a scripted web site with forms to add specific information, such as the on line reengineering bibliography and program understanding tool repositories. The underlying structure imposed by the forms also supports the possibility of structure-based searching. This structure itself can only be changed by the maintainer of the site.

In a wiki system, users can add contents as in a form-based site. Moreover, they can modify the underlying structure, which is present in the navigation structure provided by the page links created by the users. The resulting structure is less explicit (and hence less suitable for structured querying), but more flexible. Thus, wiki is particularly useful where the underlying structure is likely to be subject to change. A typical example is the development and evolution of a *taxonomy*, which aims at *finding* a given structure for a research area.

3. Reengineering Wiki Topics

The *Reengineering Wiki*⁷ is part of the larger *Program-Transformation.Org*⁸ wiki. This is a site devoted to the general area of program transformation, covering topics such as optimizations, refactorings, term rewriting, strategies, domain engineering, and generative programming. Moreover, *Program-Transformation.Org* acts as an on line annotated bibliography in the area of domain-specific languages [4]. It is also used as a documentation forum for various tools, such as XT [6] and Stratego [9], as well as a collection of grammars with associated tools called *The Online Grammar Base*.

Since reengineering shares several topics of interest with program transformation (in fact it can be viewed as a form of program transformation), it was decided to take advantage of this existing wiki and infrastructure.⁹ The entire site currently contains more than 1300 pages. The Transform web, of which the *Reengineering Wiki* is a substantial part, consists of 750 pages surveying aspects of program transformation. This part of the site currently generates 20,000 page views per month, distributed over 1000 unique visitors. The site is intended for creating *on line surveys* in several areas (such as [10]).

One category of topics includes resources of interest to the reengineering community. The *Program-Transformation.Org*

⁷See <http://www.program-transformation.org/re>

⁸See <http://www.program-transformation.org/>

⁹Based on Twiki technology: www.twiki.org.

wiki includes separate lists of conferences, workshops, journals, research groups, people, companies, and tools devoted to program transformation. The list of conferences is sorted by date, helping to plan paper submissions.

The backbone of the *Reengineering Wiki* is formed by a taxonomy of reengineering topics. The taxonomy used is a modernization of the classic taxonomy given by [2], which covers such topics as forward and reverse engineering, restructuring, redocumentation, and design recovery. Extensions of the taxonomy include topics such as program comprehension, architecture extraction, software evolution, and software maintenance. The resulting taxonomy is a graph that is available through the navigation structure.

For several of the topics, more detailed pages or subtopics have been created. The choice of these topics largely depends on the visitors of the site: any one can add or edit topics he or she likes. One of the topics that is covered in significant detail is *decompilation*: the authors¹⁰ of the original decompilation pages have moved their contents to the *Reengineering Wiki*, in order to make it easier for others to contribute. Other more elaborate topics include exchange formats, architecture extraction, documentation generation, and software visualization.

A typical page on the *Reengineering Wiki* includes a short definition / description of the topic (often taken from the literature), references to relevant papers, a list of related topics, and a list of resources such as web sites of interests or text books, upcoming conferences, etc.

Last but not least, visitors can extend the *Reengineering Wiki* with personal contributions: it is possible to create a page devoted to yourself, or to some of the recent papers you wrote. This is particularly useful for explaining how a particular paper relates to some of the topics discussed on the *Reengineering Wiki*, or for industrial researchers whose company does not support personal home pages.

4. Invitation to Join

The *Reengineering Wiki* is an open web forum devoted to reengineering. Its content can be determined by you!

You are invited to

- Browse through the wiki in search for reengineering papers, conference calls, research groups, companies, etc;
- Add (critical) remarks to any page you (don't) like;
- Contribute information on reengineering events you are aware of;
- Use the wiki to host the homepage of an event you organize;
- Annotate pages with links to related research results;

¹⁰Most notably Cristina Cifuentes and Mike van Emmerik.

- Create a page devoted to your particular area of expertise.
- Make the *Reengineering Wiki* into a living taxonomy, by extending or modifying the underlying structure.

You are invited to visit the site, and make yourself heard!

Acknowledgments Leon Moonen made us aware of wiki in the first place. Merijn de Jonge set up the first wiki used by program-transformation.org. Eelco Dolstra helped with installing the first twiki at Utrecht. Henk Penning was helpful in hosting the program-transformation.org domain and twiki at the web server of the Institute for Information and Computing Sciences of Utrecht University. Last but not least, thanks to all those wiki users who provided content and edited pages.

References

- [1] M. G. J. van den Brand, P. Klint, and C. Verhoef. Reverse engineering and system renovation – an annotated bibliography. *ACM Software Engineering Notes*, 22(1):57–68, 1997.
- [2] E. J. Chikofsky and J. H. Cross. Reverse engineering and design recovery: A taxonomy. *IEEE Software*, 7(1):13–17, 1990.
- [3] A. van Deursen, P. Klint, and C. Verhoef. Research issues in software renovation. In J.-P. Finance, editor, *Fundamental Approaches to Software Engineering (FASE '99)*, Lecture Notes in Computer Science, pages 1–21. Springer-Verlag, 1999.
- [4] A. van Deursen, P. Klint, and J. Visser. Domain-specific languages: An annotated bibliography. *ACM SIGPLAN Notices*, 35(6):26–36, June 2000.
- [5] R. C. Holt, A. Winter, and A. Schürr. GXL: Toward a standard exchange format. In *7th Working Conference on Reverse Engineering*, pages 162–171. IEEE Computer Society Press, 2000.
- [6] M. de Jonge, E. Visser, and J. Visser. XT: a bundle of program transformation tools. In M. van de Brand and D. Parigot, editors, *Proceedings of Language Descriptions, Tools and Applications (LDTA 2001)*, volume 44 of *Electronic Notes in Theoretical Computer Science*. Elsevier Science Publishers, 2001.
- [7] S. Rugaber and L. M. Wills. Creating a research infrastructure for reengineering. In *3rd Working Conference on Reverse Engineering*, pages 120–130. IEEE Computer Society Press, 1996.
- [8] S. E. Sim and M.-A. D. Storey. A structured demonstration of program comprehension tools. In *7th Working Conference on Reverse Engineering*, pages 184–193. IEEE Computer Society Press, 2000.
- [9] E. Visser. Stratego: A language for program transformation based on rewriting strategies. System description of Stratego 0.5. In A. Middeldorp, editor, *Rewriting Techniques and Applications (RTA'01)*, volume 2051 of *Lecture Notes in Computer Science*, pages 357–361. Springer-Verlag, May 2001.
- [10] E. Visser. A survey of rewriting strategies in program transformation systems. In B. Gramlich and S. Lucas Alba, editors, *Workshop on Reduction Strategies in Rewriting and Programming (WRS'01)*, volume 57 of *Electr. Notes in Th. Comp. Sc.* Elsevier Science Publishers, 2001.