

DEEP THOUGHT

Web based System for Managing and Presentation of Research and Student Projects

Tomáš Gregar, Radka Pospíšilová and Tomáš Pítner

Department of Computer Systems and Communications, Masaryk University, Botanická 68a, Brno, Czech Republic

Keywords: Python, Subversion, Project management system, Web 2.0, Tagging, Ontology, XML, Wiki, Trac, Plug-in, Bug-tracking, Visualization, Portal, Metadata, Agile programming, Extreme programming, Semantics.

Abstract: There are plenty of projects solved each day at academic venues – small in-term students' projects without any real usability, bachelor and diploma thesis, large interdisciplinary or internationally supported projects. Each of them has its own set of requirements how to manage it. Aim of our paper is to describe these requirements, and to show how we tried to satisfy them. As a result of further analysis we designed and implemented system *Deep Thought* (under development since autumn 2007), which united the management of distinct categories of projects in one portal. System is based on open-source technology, it is modular and hence it is capable to integrate heterogeneous tools such as version control system, wiki, project presenting and managing. This paper also introduces aims of the future development of the system, such as interoperability with other management systems or better connection with the lecture content and teaching process.

1 INTRODUCTION

Our department has mixed objectives – learning and research. Programming oriented lectures are held, bachelor and diploma thesis (aimed on developing some software utility or describing some design patterns, frameworks etc.) are offered. On the other hand, also non-learning research projects are solved. Each project has to be properly managed – and its original cause and importance is irrelevant for management purposes. Every project has its program artifacts, whose have to be presented and managed, its deadlines and users.

For example, in the particular case of the department of the authors, we can talk about tens of each-term student projects, circa thirty bachelor and diploma theses and some non-learning research projects each year. This amount is growing because of rising popularity of programming lectures (such as *Introduction to Java*, *Modern tagging languages* etc.). Such amount of the projects requires formal management. Following sections show manager requirements, techniques which was adopted to solve them, and finally, the system Deep Thought itself.

2 MANAGER REQUIREMENTS

During last five years of managing student and research projects, these manager requirements were identified:

- Centralized and hierarchical repository – required for further usage and archiving. Hierarchical division is vital for preserving a overview in such broad database. Students' projects should be archived too.
- Project describability – functionality description, assignment, reference lists etc. These *metadata* are important for usability preservation in the future. Further development and employment of the project is impossible without documentation.
- Project status examining – the manager has to be informed about develop processes. Some of them are explicitly created by developer (technical notes), others are created automatically (statistics, quality metrics). Possibility of monitoring the status can be seen also as learning issue. Processes like *examination of the programming code quality* or *monitoring of development* have learning values.
- Output presentation – If the project evolves into

some usable tool, it is important to have utilities and site to present it to the possible users. Such presentation have to be user-friendly, it must offer downloadable artifacts, user documentation, bug reporting tool, some interface for communicate with the developers etc.

Further information about these issues can be found in (Gregar et al., 2008) (written in Czech).

3 OLDER APPROACH

We use system *Subversion* (SVN)(Collins-Sussman et al., 2008) for storage of students' and research projects. It is version control system, capable to upload new versions, resolve document conflicts (document conflict is a state, when exact part of the document is edited by two or more users simultaneously, with different output) etc. SVN is integrable in the majority of operating systems, into the IDEs of different programming languages. Subversion works as accessible and safe archive.

Wiki is a suitable tool in the area of creating and editing data about projects, as well as collaborative preprocessing, analysis, background research, projects reviews etc. Wiki allows to easily create and format hypertext documents. We have been using MediaWiki(Barrett, 2008)¹ implementation with some modifications (for example cooperation with authentication service from the information system of the university).

We also utilized system Maven2(J. Van Zyl et al., 2008). It is Java-based shell for automatic compiling and building an application. It provides a possibility to generate various reports and quality measurements of the project.

4 PROJECT MANAGEMENT SYSTEMS

Every category of projects has slightly different accent of the common requests (i.e. research project should has better output presentation, in-term project requires deep status examining). It would be a benefit to have a portal, which can satisfy all main requests and can be used for all categories of projects. Other important requirement was to retain wide amount of data from already managed projects. Search of such project management system started in November 2007.

¹<http://kore.fi.muni.cz:5080>

We did not find any suitable existing system – either the license was too expensive (for example portal for MIT's Simile project², Jazz from IBM) or lack some important features (for example SourceForge³, XPlanner).

System *Trac* (see section 3.1) satisfies our requirements for the managing one project but it *cannot work as a multiproject portal*. We decided to utilize and extend Trac as a part of our portal. During state of the art analysis we found several projects, one of them was named DrProject, multiproject system based on Trac. But system was functionally insufficient for our needs that time. Without finding any usable project, we decided to develop our own portal system.

4.1 Trac

Portal Deep Thought was build upon the system Trac (Edgewall Inc., 2008). Trac is an enhanced wiki and issue tracking system for one project software development. Trac uses a minimalistic approach to web-based software project management. It has not all abilities we require, but it has a modular structure and can be modified (added other data interfaces, user rights processing, new modules for new functionality etc.). Trac is written in object oriented script language Python.

It has these standard modules:

- User friendly browser of the project code. SVN repository is one of the supported tools. Browser also allows viewing source code as differences between distinctive versions, the antecedent revisions of the code etc.
- Bug reporting and tracking utility – structure of the page concerning this issue, named *ticket*, can be modified according to author's requirements.
- Project management – definition of the development milestones, project component declaration, development monitoring etc.
- Presentation materials creation with integrated Wiki system.

Trac can manage only one project. Other problems comprise a requirement to have access to server command line (for creating project, updating system), lack of the free-form GUI modification etc. It also does not allow localization in any way (which is problem in non-english speaking countries.)

²<http://simile.mit.edu/>

³<http://sourceforge.net>

4.2 DrProject

DrProject is based on similar idea as our implementation. Both of them are derived from Trac and specialized for academic purposes. Massive changes in Trac's source code had been done due to functionality extension in DrProject. It causes inability to use some plugins. It has no internationalization support.

On the other hand, DrProject in the recent time grows into useful application. It has advanced tagging engine (relation between tag and its project visualization, global vs. local tagcloud). System has also nice, javascript framework driven, GUI.

5 DEEP THOUGHT

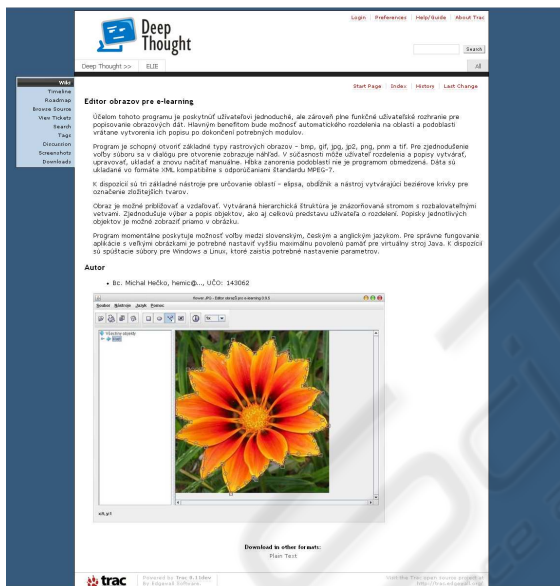


Figure 1: Deep Thought screenshot - project presentation.

5.1 Requirements and Development

Design and implementation of the system are done by prototyping using extreme programming techniques. Wiki was used in analyzing important capabilities and for communication about the development process. We try to unify the project management as much as possible while covering the wide range of requirements coming from very different projects' nature. In the analysis and development of such project management system, we took the advantage of a rich built-in functionality (in Trac) but it had to be strongly modified to satisfy the important functionality requirements. Modifications contain:

5.1.1 Multiproject Management

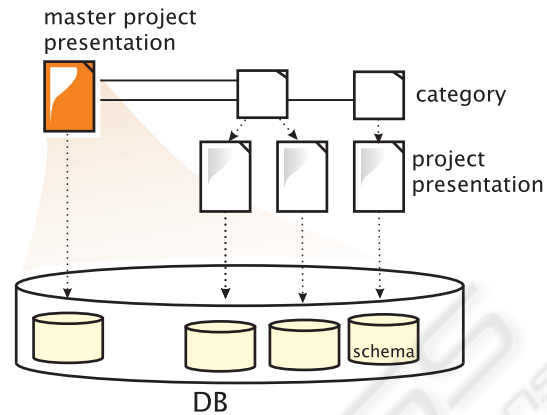


Figure 2: Deep Thought Structure.

In Trac, every single project, configuration, source codes, user rights are treated separately. Trac programmers community generated some concepts how more projects can be treated (none of them officially accepted). Some proposals change database structure, others base system components. We would like to avoid such changes. They decrease a possibility to use third-party modules and plug-ins without complex changes of the source code.

We found one elegant solution⁴. It is based on utilization of schemes in shared PostgreSQL database. This minimalizes interventions in source codes and allows to gain global view (with help of user defined database stored procedures) on the data.

Other important issue is to centralize the settings. Configuration files used by the system work with a concept of inheritance. The whole project file-system was rebuilt into multi-project structure, in which is global configuration separated from particular project information. Access rights management have to be centralized as well. Plug-in *TracForge* was used to do this. Authentication proceeded by HTTP-Basic, was changed to utilization of faculty Kerberos server.

5.1.2 Project Presentation

Presentation of the projects is also simplified by hierarchical structure of project categories (Pospisilova and Gregar, 2008). User can switch between projects and categories. Management of these categories and their content is easily done by the system administration module. It uses wiki pages to create this hierarchical structure.

New database and directory structure is created within Deep Thought installation. This structures are

⁴<http://trac.edgewall.org/wiki/TracMultipleProjects/ComprehensiveSolution>

ready to add further projects. Meta-project, called *master*, is also created. It does not describe any particular project. It contains texts and documents about the entire portal, shows global ticket lists, global tag lists, it allows to administrate the whole system etc.

5.1.3 Project Management Simplification

Deep Thought system has *New project* web administration module (in the Trac system it had to be created in command line and set manually). With new project creation, the path to the repository is mapped, basic web-pages are created, user rights for basic user-groups are adopted from master settings. Transportation of projects, which were managed with the older approach, became easier.

5.1.4 Internationalization

Python allows to develop internationalized programs. Trac does not contain such feature. One of its development branches tries to add this support. We used it as a basis for our work. Many changes had to be made to clarify, clean up and correct that branch. We also redefined a set of translation strings. That capability is important (see section 2), but no other multiproject system utilizes it. We have started to solve localization of wiki pages and tickets after we finished localization of texts and the other GUI features. We need variants for distinct languages in the internationalized DT. Unfortunately, the modification of wiki version system is relatively complicated – it requires many changes in the Trac core and in the GUI. Therefore we decided to create a new wiki-macro. Tag lang, occurred in the page, signalize that the latter text is in the defined language. Tag contains locale information about consequential text (see an example in screenshot).

5.1.5 Better Output Presentation for Users

Presentation functionality was enriched by external Trac plug-ins, sometimes modified to offer better incorporation or better functionality (A. Thomas et al., 2008).

- Pictures and Screen-shots gallery – ScreenshotsPlugin⁵ module
- Databases of downloadable files – DownloadsPlugin⁶ module
- Incorporation of output from Maven tool – slightly modified MavenTrac module

⁵<http://trac-hacks.org/wiki/ScreenshotsPlugin>

⁶<http://trac-hacks.org/wiki/DownloadsPlugin>

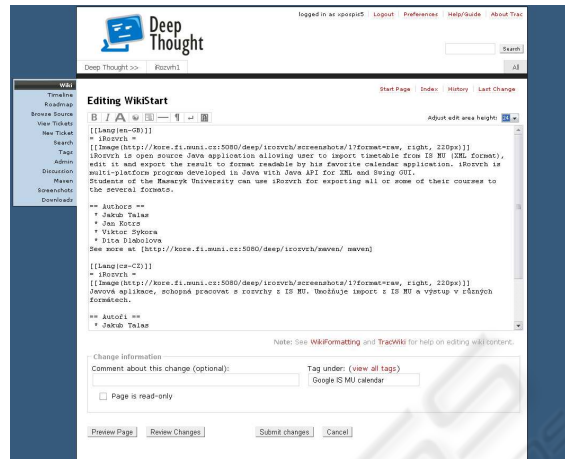


Figure 3: Multilanguage wiki-page editing.

- Discussion forum
- Tagging – user-driven tagging of documents within system. This plug-in – TagsPlugin – was modified to provide global overview of the tags. It will be described in next section.

5.1.6 Design Changes

Trac offers some possibility to change a design. The basic configurable part of the design (like logo) are adjustable in the project configuration file. Output of the system is written in modular HTML. Slight change, creating the global container, had to be made to enable *page* design. Graphic changes have been made by cascading styles (CSS).

6 SEARCHING AND SEMANTICS

The important part of any presentation portal is searching. Deep Thought uses two different ways to find a document, and third way is under the development:

- Fulltext search
- Tagging – It employs usage of user-defined tags (i.e. words or multi-word concepts) connected with documents (i.e. wikipages and tickets). These tags build lightweight database, in which can be searched. Such database also allows to create visual presentation, which shows what concept is more common, searched more etc. It adds another web 2.0 capability to the system (Golder and Huberman, 2006).
- Advanced semantic search – Under development since November 2008. Subproject, solved as a

diploma thesis, is aimed to incorporate semantic approach into document processing. It should be done by modifying the text editor module. Tagging of the text with concepts defined by ontology, proposing them to user automatically, or even tag the document automatically should be possible with such modifications. This *tagging* can produce better searching output than user-driven tagging because of formal description of concepts. Possibility to use of other semantic-driven techniques (visualizations, links to semantic dictionaries etc.) belongs also among other benefits of semantic definition of document content. This subproject will collaborate with RDFS-repository Sesame 2 (Aduna Inc., 2008).

7 FUTURE GOALS

System is online, projects from our archive were included in the portal, their sites were set up. Portal gives us a possibility to present already finished projects, without any management processes, as well to start new project with full management tools utilization. Further goals in the near future comprises:

- Source code and internationalization actualization – Trac, which is base of our portal, is active project and its source code evolved since it has been used in the DT. Some attributes and functions of its core were improved. Hence transport of DT to the actual build of the Trac system would be an improvement. Such transportation is not easy, because of numerous changes in source code and implemented internationalization.
- User rights system modification for better usage outside our faculty – new set of automatically generated hierarchical rules have to be prepared. These rules will be shared among the whole system, every project, and will build multilevel hierarchy of user rights. Other goal in this area is to allow to keep some documents (like wiki pages) private.
- Further GUI improvement – alternative graphic design will be created and system will be modified to change designs easily. Support for including the multimedia data (such as mind-maps, flash) will be add. Higher rate of multimedia usage is caused by utilizing the system for e-learning.
- Incorporation of semantic-related technologies – to implement new plugin as described in *Searching* section). Adding automatic relations between pictures and text.

8 CONCLUSIONS

Preparation and managing of large amount of research and in-term students' projects is challenging and requires formal processing. The introduced tool unites (without big issues) the older distinctive processes for different categories of projects and adds new qualities. The system structure and the development process enable further modifications, like Trac plugins incorporation or new add-ons development.

The paper provides basic overview of Deep Thought Portal, reasons this system was created, its structure, characteristics, implemented modules. The paper also briefly introduces the possibilities of expanding the functionality of the system and describes basic functions for each declared type of users. It also describes its cooperation with semantic technologies and intended usage this advanced functionality in the searching and enrichment of the stored project data and metadata.

System Deep Thought can be used in any academic venue, where bigger amount of developer projects have to be managed.

REFERENCES

- A. Thomas et al. (2008). Trac hacks - plugins macros etc. <http://trac-hacks.org>.
- Aduna Inc. (2008). Openrdf.org, sesame documentation. <http://www.openrdf.org/documentation.jsp>.
- Barrett, D. J. (2008). *MediaWiki*. O'Reilly Media, Sebastopol, 1st edition.
- Collins-Sussman, B., Fitzpatrick, B. W., and Pilato, C. M. (2008). *Version Control with Subversion*. O'Reilly Media, Sebastopol, 2nd edition.
- Edgewall Inc. (2008). The trac project. <http://trac.edgewall.org/>.
- Golder, S. and Huberman, B. (2006). Usage patterns of collaborative tagging systems. In *Journal of Information Science*. SAGE Publications.
- Gregar, T., a, R. P., and Pitner, T. (2008). Sprava vice projektu a prezentace jejich vysledku. In *Sbornik konference Mekon 2008*. VSB - Technicka univerzita.
- J. Van Zyl et al. (2008). *Maven: The Definitive Guide*. O'Reilly Media, Sebastopol, 1st edition.
- Pospisilova, R. and Gregar, T. (2008). Deep thought. <http://kore.fi.muni.cz:5080/deep/master>.