KiWi - Knowledge in a Wiki

Sebastian Schaffert[1], Julia Eder[1], Matthias Samwald[2], Andreas Blumauer[2],

[1] Salzburg Research Forschungsgesellschaft, Jakob Haringer Str. 5/II, 5020 Salzburg, Austria
sebastian.schaffert@salzburgresearch.at
[2] Semantic Web Company, Lerchenfelder Guertel 43,
1160 Vienna, Austria
{m.samwald, a.blumauer}@semantic-web.at

Abstract. The objective of the project KiWi is to develop an advanced knowledge management system (the "KiWi system") based on a semantic wiki. This poster describes the KiWi project, its technical approach, goals and the two use-cases which will be covered by the KiWi-System.

Keywords: semantic wiki, project knowledge management, software knowledge management, reason maintenance, personalization, information extraction

1 Introduction

Knowledge management in software development and project management is an exciting problem, as it involves tacit knowledge (e.g. about processes), distributed knowledge (different people, different systems), and many different kinds of semantically rich content (e.g. source code, documentation, tutorials, project work plans) that is strongly connected on the conceptual level.

Current knowledge management systems only insufficiently support knowledge management in such areas, as they are not flexible enough to handle and integrate these kinds of content and provide only insufficient support for tacit knowledge.

In order to go beyond the functionality of current systems, existing technology, particularly in the areas of knowledge representation, visualisation, reasoning and reason maintenance, information extraction, and personalisation, is not mature enough for immediate deployment as it lacks performance, maturity and usability. Significant progress and effort in these areas is necessary in order to create advanced semantic wikis that support the user in the ways outlined above and as required in our use cases.

Unfortunately, traditional wiki systems do not currently provide real support for knowledge management tasks: Firstly, current wiki systems are not really able to work with non-textual content like spreadsheets, diagrams or video material. Secondly, wikis are good at editing and storing content, but provide no or very limited support for storing and processing computationally "interesting" data, like project resources, calendar dates, etc. Thirdly, wikis do not provide support for the semantically "rich" content that participates in most knowledge management tasks.

The rationale for developing the KIWI system based on a semantic wiki is that – as demonstrated in our two use cases – organisations in knowledge-intensive areas are nowadays more and more using wikis and other social software systems for their knowledge management tasks like knowledge sharing (e.g. ideas, references), collective knowledge creation (e.g. architecture, documentation), and project coordination (e.g. project resources, work plan, bug tracking).

2 Use Cases

The development of the KIWI system will be accompanied by two concrete use cases that address current and real-world problems of our two industrial partners, Sun Microsystems and Logica, in the areas of software knowledge management and project knowledge management. While the use cases will address concrete knowledge management issues at the two companies, they are of generic nature and representative of large classes of applications.

The main show case for the KIWI system within Sun will be the NetBeans engineering organization, which does end-to-end development of the NetBeans Integrated Development Environment. Since the development is done in an open-source fashion, it involves many teams both from inside and outside of the company. This use case is mainly concerned about the management of knowledge about e.g. software components, software competencies, documentation, etc

The second use case addresses development tasks in Logica, which require many different skills and competencies. The effective sharing of knowledge in and between projects is a complex undertaking both technically and organizationally. The challenge with knowledge sharing in Logica is generally that substantial knowledge is built in a development project and currently only survives in the minds of the developers and the project managers; thus the exploitation of knowledge is limited to personal experiences. This use case is concerned about the management about project relevant knowledge, e.g. work flows, resources, teams, meetings, schedules, etc