salzburg research

KIWI - Knowledge in a Wiki

FP7 Information Days
Luxemburg, 13. December 2007

Dr. Sebastian Schaffert
Salzburg Research Forschungsgesellschaft
Salzburg NewMediaLab
sebastian.schaffert@salzburgresearch.at

SNML salzburg newmedialab

Outline

- Project Vision & Objectives
  - what the project will do (briefly)
- Success Factors
  - or: why I believe KIWI was successful
- Challenges
  - what the project will not do but would benefit of

Objectives

Motivation

- Knowledge management involves
  - many different kinds of "rich" content (plain text, documents, images, audio/video, source code, discussion protocols, ...)
  - different user- and domain-dependant workflows,
  - sharing and collaboration between users
- Wikis are nowadays increasingly used to support knowledge management as
  - they allow simple sharing and collaboration (usually text only)
  - they are easy to use (syntax, linking, versioning)
  - they don't impose a predefined workflow on users ("dictate of the system")
  - **but:** they also provide virtually no support for the user, all processes are managed by social convention, presentation is "just" text

Objectives

1. to develop an **"enhanced wiki vision"** (KIWI vision) describing how "convention over configuration" combined with semantic technologies help to build knowledge management systems that *support* the user, but do not *restrict* the user

Objectives

2. to develop a **"collaborative web-based prototype system"** (KIWI system) realising this vision by extending the semantic wiki IkeWiki with the following functionalities:
   - support for non-rigid work flows and processes
   - advanced interactive visualisation and editing
   - improved knowledge representation
   - enhanced reasoning and querying capabilities
   - reason maintenance
   - information extraction
   - personalisation

Objectives

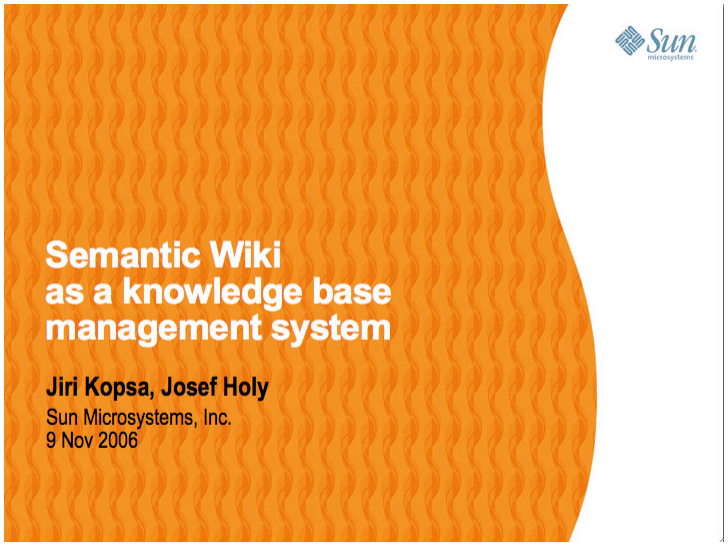
3. to write a "**KIWI handbook**" summarising project outcomes and documenting the functionalities of the KIWI system; the handbook will contain:
   - the KIWI vision
   - user and developer documentation for the KIWI system
   - descriptions of the use cases
   - best practices for using the KIWI system for knowledge management purposes

Use Case 1: Software Knowledge Management

- knowledge in and about software systems involves many different kinds of (rich) content:
  - source code
  - documentation
  - minutes of group meetings
  - bug reports,
  - ...
- development of Sun's Netbeans IDE is developed by a large community with a team of 150 core developers employed by Sun Prague
- knowledge gathered and developed during the software development process is distributed over many different kinds of systems (wikis, bug trackers, blogs, discussion forums, mailing lists)

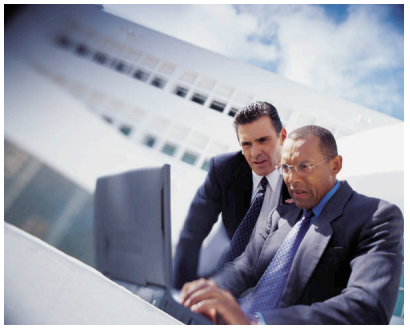
Use Case 1: Software Knowledge Management

- goal of the use case is to create a semantic wiki system that allows to capture software development knowledge, integrate different kinds of content, allows to semantically query for content, provide different kinds of views on the content, etc (see also Jiri Kopsa's slides if needed)



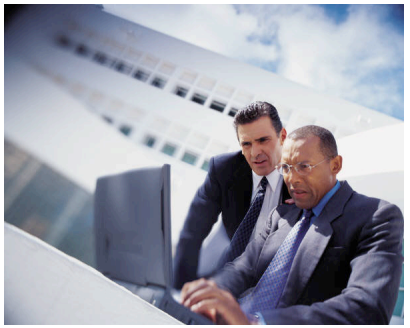
Use Case 2: Project Knowledge Management

- WM-data is a IT and management consultancy, particularly for the Navision (MS Dynamics NAV) ERP system
- consultancy is project based and specific for every customer
- nonetheless, much of the project knowledge is similar between different projects and customers
- much of the project knowledge is tacit and currently lies in the heads of experienced project managers



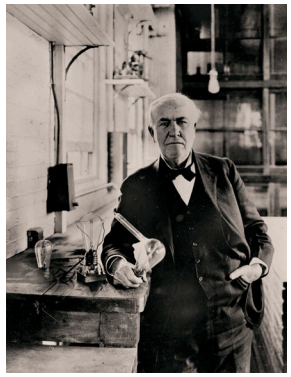
Use Case 2: Project Knowledge Management

- goal of the use case is to create a semantic wiki based tool that allows:
  - knowledge sharing in, and between, distributed development projects by supporting distributed wiki systems, different kinds of content, and easy sharing and access to knowledge
  - knowledge sharing in software process improvement by capturing tacit knowledge in formal models and using advanced visualisations to present complex sets of knowledge items
  - knowledge sharing in customer-oriented high-performance projects by supporting a close cooperation between developers and customers



Success Factors

What I believe made KIWI a success
(Scientific & Technological Content)



- KIWI has a concrete, graspable, clearly defined outcome
  - at the end of the project, we will have a single, demonstrable system that actually implements what we promise
  - theory touching the ground!
- KIWI addresses real-world problems identified by its industrial partners
  - no blue sky research, application oriented
- KIWI improves already existing technology where it has deficiencies
  - IkeWiki
- KIWI builds upon previous EU and national projects and takes their outcomes to the next level
  - REWERSE, QVIZ, Dynamont

What I believe made KIWI a success
(Project Setup)

- The KIWI consortium is well balanced
  - complementary expertise, exactly as needed for the project
  - 3 universities, 1 research centre, 3 industrial partners (1 SME)
  - 4 countries (Austria, Germany, Czech Republic, Denmark)
- Significant effort goes into Dissemination & Demonstration
  - 64 out of 388 PM (= 18,5%)
  - the importance of marketing research results is often underestimated
- The project workplan and budget was planned as if it was already a description of work
  - thorough and realistic calculation

What I believe made KIWI a success
(Social Dynamics)

- We had a proposal kick-off meeting with all partners 2 months before the submission deadline
  - people learn to know and trust each other
  - a rough project workplan was set up (Gantt diagram)
  - social bonds are important!
- I tried to collect issues and send out email to the partners only once a week ("KIWI weekly update")
  - no email overload at the partners (be nice to them!)
  - clear definition of who is supposed to do what by which date (make it easy for them!)
- We communicated often via Skype to clarify issues and tasks
  - keep in touch with them (but not by email)
- We signed Letters of Intent as part of the proposal
  - demonstrates commitment
  - makes management aware of the project

What I believe made KIWI a success (Formalia)

- We made use of the pre-proposal check service offered by the Commission, which gave valuable feedback
- We tried to take into account *all* available background material:
  - work programme and guide for proposers
  - background material specific to the unit/strategic objective
  - slides of the Call 1 Information Days
  - general EU policy documents

Challenges

Challenges KIWI won't solve
(but it would profit from solutions!)

- efficient, distributed content and knowledge repository
  - with reasoning support (rule-based as well as other)
  - with support for both content and metadata
  - **and** with support for billions of information items (content, triples, ...)
  - KIWI will only provide a "small and specific" solution!
- non-textual content
  - (semi-)automatic metadata extraction
  - annotation
  - breaking the "format lock"
- and of course lots of other topics that are outside the direct project scope (e.g. other Semantic Social Software) ...

Contact

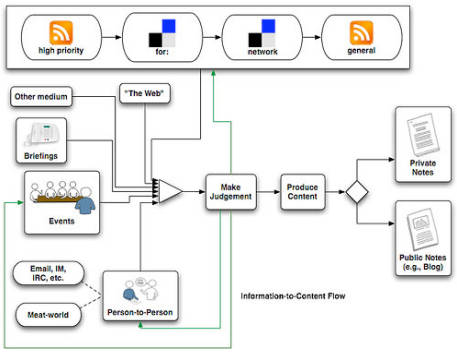
| Dr. Sebastian Schaffert
| Salzburg Research Forschungsgesellschaft
| Jakob Haringer Str. 5/II
| A-5020 Salzburg

| sebastian.schaffert@salzburgresearch.at

Appendix

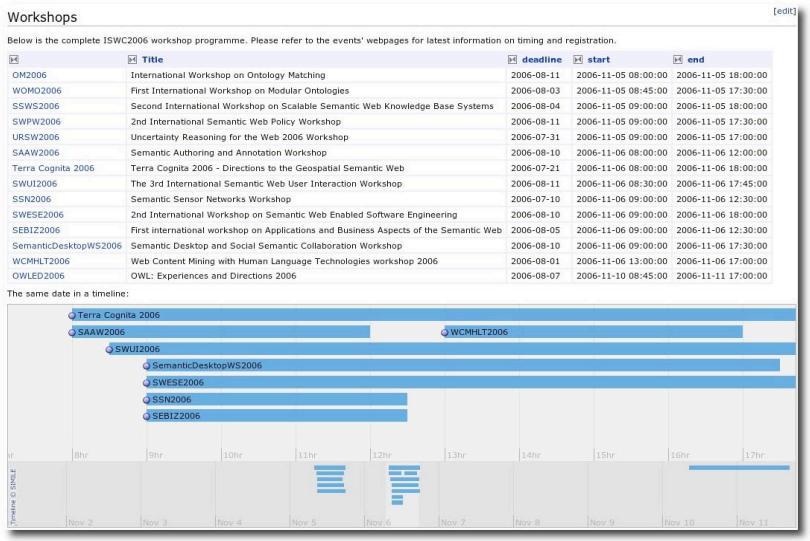
Support for non-rigid work flows and processes

- current content/document/asset/knowledge management system impose rigid work flows on how knowledge sharing takes place (e.g.: submit -> review -> correct -> publish)
- such work flows are often barriers for sharing knowledge
- wikis are successful because they do not impose work flows; social convention takes over
- Semantic Web technologies allow to offer the advantages of work flows when they are followed, but also give the flexibility to deviate when needed
- **goal**: develop support for such "non-rigid work flows" through appropriate work flow models and visualisations



Advanced interactive visualisation and editing

- different domains have different ways of presenting knowledge
- visualisations are often superior to plain text (as in wikis)
- Semantic Wikis have the potential to automatically generate domain-specific and generic visualisations out of the knowledge base
- same argument for editing: for a project manager, it is more suitable to insert project data using e.g. a Gantt chart or a resource table than doing generic metadata annotations
- **goal**: develop a generic frame-work for visualisation and editing that does not require hard-coding in the sourcecode of the semantic wiki system



Improved knowledge representation

- Current knowledge representation in IkeWiki is "ad hoc", based on Jena and PostgreSQL
  - no "real" reasoning support, and what is there has bad performance and wrong goals (validation of knowledgebase is not really useful for a wiki)
  - no versioning of meta-data (but this is required by users, otherwise they won't trust a "semantic wiki")
  - no integration of different, distributed knowledge bases (but this is a requirement of the use cases)
  - no support for annotating only parts of pages/images/other objects
  - ⇒ not well prepared for enabling technologies!
- **goal**: to develop a revised knowledge representation component that addresses these issues

Enhanced reasoning and querying capabilities; reason maintenance

- semantic wikis, particularly with advanced visualisations, personalisation, etc, require different support for reasoning and querying than is currently offered by e.g. Jena, Sesame, etc
  - OWL(-Full/-DL/-Light) is nice for many tasks, but completely fails for rule-based derivations; these are, however, typical for most kinds of applications, and arguably represent the "interesting" part of reasoning
  - current reasoners for the Semantic Web do not really take into account reason maintenance; this is, however, a prerequisite for efficient versioning of meta-data in a reasoning system, and can also increase the "experienced performance" and responsiveness of the system
- **goals**:
  - to develop a rule-based language that can be used by users of the wiki to query and to specify derivation (and possible action) rules that are capable to query content and knowledge base in a unified fashion; also, a simple and intuitive way to specify such queries and rules is sought for
  - to develop a reason maintenance component for this language that gives users a way to understand why certain derivations exist, that allows versioning of updates to the knowledge base, and that allows easy updates

Information extraction

- current semantic wikis provide no support for annotating content (besides providing an easy-to-use interface to enter annotations)
- content of wiki pages usually already contains much of the knowledge, albeit in natural language text
- annotation is for many users a very daunting task, because it requires understanding of the underlying knowledge models and concepts behind them
- existing natural language technologies can partly extract meta-data out of the natural language text and use this information to interactively guide users in the annotation process (e.g. by providing custom wizards, or even by simple reordering of the offered concepts)
- **goals**:
  - to develop an information extraction component that semi-automatically extracts meta-data out of wiki pages to interactively guide the user through the annotation task

Personalisation

- different users have different roles in knowledge management processes (e.g. developer, tester, documentation writer, customer)
- different users might also have different tasks and/or preferences
- in both cases, personalised presentation of content and user interface provides support for the user
- a semantic wiki can support this by storing user/group models in the knowledge base, by offering a reasoning component, and by offering advanced visualisations and editors that implement personalised access to the wiki
- **goals**:
  - develop appropriate user and group models for representing properties and preferences of groups wrt. the KIWI system
  - automatically track the usage of single users and user groups in order to refine user and group models
  - develop rules that dynamically adapt the presentation of the content and user interface to the user