Towards a Collaborative Semantic Wiki-based Approach to IT Service Management

Frank Kleiner  
(FZI Forschungszentrum Informatik an der Universität Karlsruhe (TH)  
kleiner@fzi.de)

Andreas Abecker  
(FZI Forschungszentrum Informatik an der Universität Karlsruhe (TH)  
abecker@fzi.de)

Abstract: This paper proposes a collaborative approach to aspects of IT service management (ITSM) for agile information technology landscapes composed of technically knowledgeable users and fast changing requirements. The cost-intensive and time-consuming tasks of building and maintaining a configuration management database (CMDB) are accomplished in a distributed, collaborative way in a semantic wiki in cooperation between administrators, service desk personnel and users of the systems. Moreover, the paper introduces a flexible, wiki-based approach to change management for agile environments. Automatic consistency checks based on semantic information extracted from the wiki prevent misconfigurations and the unintentional application of conflicting configuration changes.

Key Words: semantic wiki, collaboration, ITSM, configuration management, CMDB

Category: H.5.3, I.2.4, K.6

1 Introduction

IT service management (ITSM) is the discipline which is responsible for dealing with all information technology processes which are relevant for successfully reaching a company’s business goals. In contrast to IT management disciplines of the past, which dealt mostly with technical aspects of information technology, ITSM provides a customer-centric view, which focuses on the contribution of IT to the customer’s business success. There are several frameworks, which give guidelines for implementing ITSM processes, with the Information Technology Infrastructure Library (ITIL) being the most prominent one [Clacy and Jennings 07].

Configuration management [Lacy and Macfarlane (07)] is the process within IT service management which is responsible for describing all entities, which are used to provide IT services, as well as the relationships between these entities. Entities relevant for configuration management are referred to as configuration items (CIs). Descriptions of configuration items and relationships between configuration items are stored in the configuration management database (CMDB),
which is the logical abstraction of all physical databases which contain information that is relevant to the configuration management discipline within an organization [Köhler (07)]. While there exists a variety of tools for automatically populating and updating CMDBs, not all information can be determined and put in context automatically.

Wikis are a special form of web sites, in which users are able to not only read published content but are also able to add new information as well as to change existing information. While the Wikipedia encyclopedia\(^1\) is the most prominent example of a wiki, wikis are often used as collaboration tools in companies or projects in order to facilitate knowledge management between coworkers. The MediaWiki\(^2\) software, which is used to run Wikipedia, is a freely available tool which is used for setting up wiki sites in many companies. Due to its extensibility and the availability of 3rd party extensions, MediaWiki is used as the basis for the solution proposed in this paper. Changes applied to wiki articles are stored with the editing user’s name or IP address, which provides accountability. Version management allows to view differences between two versions of an article and to undo unwanted changes applied by rogue users. Talk pages are used to discuss the content of articles in order to find a generally agreed formulation. Basic security mechanisms allow to group users and assign access rights [Barrett (08)].

Semantic wikis add semantic features to wiki software. By adding explicit semantic statements [Staab and Studer (03)], wiki articles and their relation are able to be better processed by computers. Semantic wikis provide means to add relations and attributes to wiki articles. Relations in semantic wikis are used to give meaning to the links between articles, for example “London is Capital of United Kingdom” [Krötzsch et al. 06]. The freely available extensions Semantic MediaWiki (SMW) [SMW 08] and SMW+\(^3\) are the implementations of semantic wikis used in this work. SMW+ provides additional features over SMW in the area of usability. Examples are autocompletion, a semantic toolbar, which simplifies text annotation, as well as an ontology browser for visualizing categories, instances and properties [Pfisterer et al. 08].

The paper is organized as follows: Section 2 describes the implementation of the ITSM wiki. In section 3, two problem scenarios and solutions for these problems are outlined. It is described, how configuration changes as well as conflicts between installed applications can be handled with an agile approach in a semantic wiki. After a discussion of related work in section 4, section 5 gives a conclusion and an outlook on future work.

\(^1\) http://www.wikipedia.org
\(^2\) http://www.mediawiki.org/wiki/MediaWiki
\(^3\) http://wiki.ontoprise.com
2 Implementation

The paper looks at the requirements for IT processes and procedures in small and medium-sized enterprises (SME) in the IT sector as well as in research groups at universities. When looking at researchers and developers, it can be observed that there exists a high demand for the agile application of changes to services as well as to hardware and software configurations. IT service management frameworks, e.g., ITIL, have a well-established procedure for requesting, testing, documenting and applying changes in order to ensure the compliance with service requirements. The time needed for following this kind of procedures is, however, hardly accepted by developers or researchers who, for example, want to instantly share a currently developed web application with project partners or set up a demo application. While this kind of services can be considered out of the scope of classical IT service management, it plays a significant role in research groups and amongst developers. When looking at the requirements of developers and researchers, there is a need for applying changes faster and more agile than it would be possible when precisely following change management procedures described in the ITIL framework. Moreover, some changes to development and testing systems are applied directly by the user of the systems, which makes it hard for the IT department to track and document these changes. The approach presented in this paper uses semantic wiki software as the basis for creating a platform, which allows users to participate in the documentation of configuration items and their relations to each other as well as best practices for the use of IT components. In order to convince users to participate in the wiki, incentives are provided in the form of recommendations for interesting software or hardware and notifications about conflicting configurations.

Fig. 1 shows a screenshot of a wiki article describing a configuration item. The upper part of the screenshot shows the text section of the wiki article, which contains free text as well as semantic relations and attributes expressed in the Semantic MediaWiki syntax. Facts, which are extracted from semantic statements given in the wiki article, are shown in the fact box found at the bottom of the screenshot. The fact box is used to summarize all facts found in a wiki article and allows to easily export all facts contained in a wiki article as RDF feed for the use in external applications. Facts formulate an attribute-value or a relationship-value statement. Relations describe which instances are linked to which other instances. For example, the relation “is connected to” shows that the notebook described in the wiki article is connected to a screen and a network plug. By clicking on the links representing these configuration items, their wiki articles are displayed. The “has IP address” relation is an example for a relation which links one individual to more than one other individual (n-ary relation [Noy and Rector 06]). The first part of the relation states that the notebook has the IP address “192.168.20.134”, which is described in the corresponding wiki
The data model used in the IT service management wiki is an ontology [Staab and Studer (03)]. An ontology is a formal representation of classes and the relationships (relations) between these classes. A configuration item can have one or more categories. Categories are equivalent to classes in the underlying ontology. Classes are arranged in a class hierarchy. In the example shown in the screenshot, the computer is an instance of the class “Notebook”. The notebook class is a subclass of the “Computer” class, which on its part is a subclass of the “Hardware” class.

Figure 1: Screenshot of a Configuration Item Represented in the Wiki
Information can be added and edited in two ways. The first one is the traditional way of using Wikitext [Wikipedia 09], a special syntax which is used for formatting wiki content. The second one is by using Semantic Forms [MediaWiki 08c], which help to simplify adding and editing of structured information by providing a form-based mechanism.

3 Problem Scenarios and Proposed Solutions

For illustrating some of the problems found in applying ITIL recommendations to the management of systems in an agile environment, the authors present two problem scenarios and propose solutions for the problems. The scenarios are examples in which the use of the proposed collaborative approach to IT service management would bring benefits for users as well as for service desk personnel and system administrators responsible for installing, configuring and maintaining the services and systems. The first scenario describes the problems encountered when dealing with configuration changes. In the second scenario, it is shown how conflicts and unwanted interactions between software applications installed by the users of computer systems or the installation of hardware components can lead to avoidable problems and support requests.

3.1 Configuration Changes

Problem Description: Change management and the documentation of configurations in the configuration management database (CMDB) within the configuration management process are considered essential parts when implementing ITIL recommendations. The change management process makes sure that all stakeholders are informed of pending changes and that responsibilities and priorities are assigned to changes. Implementing a back-out plan makes sure that changes are reversible in case of problems caused by the change [Köhler (07)]. The change management process consists of the following steps as described in [Köhler (07)]:

1. Need for a change is determined by administrator, service desk or customer
2. Request for change (RFC) is sent to change manager
3. RFC is evaluated; if determined to be urgent, a higher priority is assigned
4. RFC is discussed in the change advisory board (CAB)
5. Solution is implemented, tested, approved and applied
6. CMDB is being updated
When looking at the formal process for applying changes, it can be seen that it generates too much overhead and inflexibility to be of use in an agile environment. Agile environments are characterized by fast changing requirements and fast deployment of services in order to meet the requirements. Moreover, services have to be deployed by developers without a formal change process in order to provide access to recent versions of software to testers and project partners. On the other hand, there exists a need for having a full documentation of all configuration changes in order to enable system administrators to track down and fix problems.

**Proposed Solution:** The authors’ proposed solution to the problem of requesting, approving and documenting configuration changes makes use of several features of semantic wiki software. The steps for having a change applied in an organization which supports wiki-enabled agile change management are as follows:

1. Need for a change is determined
2. Change is documented in the wiki and automatic notifications are sent to stakeholders about the changes to the wiki articles – changes are approved by virtual CAB (vCAB) by not objecting
3. Solution is implemented and tested

As can be seen, the proposed solution greatly simplifies the tasks of application, approval and documentation of changes by simplifying the change management process. The three process steps are all conducted by either administrators, service desk personnel or customers who have access to the wiki and are able to perform changes to hosts or services. The automatic notifications to stakeholders are sent via the watchlist feature of MediaWiki. Watchlists allow wiki users to register for being notified when wiki articles are changed by other users. Notifications about changed articles on a user’s watchlist are displayed in the wiki and sent via email [Barrett (08)]. By registering for articles about configuration items for which administrators or customers are responsible, it is made sure that notifications about changes are sent to all stakeholders. According to traditional change management procedures, it is the role of the change advisory board (CAB) to review proposed changes before implementation. The agile change management procedure replaces the CAB with a virtual CAB (vCAB), which consists of all persons monitoring an article for changes via the watchlist function. The formal approval process of traditional change management is replaced by a soft approval process which informs members of the vCAB of imminent changes to which vCAB members can veto if the changes would bring undesired consequences. This soft approval process shortens the time needed for having a change approved while giving the possibility to prevent the conduction of undesired changes to interested stakeholders.
The MediaWiki right system is used for restricting access to articles for certain groups of users. While MediaWiki does not provide a sophisticated rights management system in its current state [MediaWiki 08a, MediaWiki 08b], the rights management suffices for collaborative environments. Wiki articles which describe configuration items which should not be changed by a certain group of users are protected from changes by the MediaWiki protect mechanism. E.g., wiki articles about servers which provide infrastructure services (mail, domain controllers etc.) are protected from changes from customers. When a customer needs a change to a server or service whose wiki article is protected, the process for having the change applied follows the traditional change approval procedure. Changes to articles, for which a customer has the rights to edit, can be applied after documenting the changes in the wiki. Discussions among administrators, service desk personnel, change advisory board members and customers can be a time consuming and error-prone task. In some instances, it is not possible to have a meeting in which all stakeholders are able to take part or it is not even known who is a stakeholder for a certain change or configuration item. By using wiki talk pages, problems with configuration items or with proposed changes can be discussed within the wiki. In the Wikipedia encyclopedia, talk pages play a significant role in the resolution of conflicts. Other important roles are in the area of coordination, e.g., discussing which articles need reworking [Viégas et al. 07]. Talk pages in the ITSM wiki play an extended role by not only discussing the content of a wiki article but also of the underlying real world entities, e.g., services or hardware components represented by the wiki article. The wiki versioning system stores all versions of the talk pages so all information discussed in talk pages is retained. The history function of MediaWiki is used to store all versions of articles which describe configuration items. With this feature, it is possible to track all changes and to compare different versions of a configuration item which is necessary for troubleshooting and accountability.

3.2 Conflicts Between Installed Applications

Problem Description: Today’s computer systems comprise a vast amount of applications and services. While most of these applications coexist peacefully, there is a number of applications that cause problems if installed or run at the same host at the same time. Examples are different versions of the same application or services which change aspects of the underlying operating system, e.g., anti-virus software or personal firewall software. Installing conflicting applications can lead to minor problems at best, up to systems not able to boot at worst. While in centrally managed environments there exists, according to ITIL recommendations, a process for testing applications for possible conflicts, and only deploying applications that are conflict-free, this is not always possible in an agile environment. In an agile environment, developers and researchers install
applications for testing purposes and for keeping on top of current developments in the software landscape. A centrally managed testing process which would definitely rule out all possible problems would impose a time delay which in most cases would be too big to tolerate. Furthermore, sometimes there exist incompatibilities between hardware components, e.g., a certain kind of notebook can be incompatible with a certain kind of projector. The use of shared resources is another area with potential for conflict. For example, if more than one wireless network is using the same frequencies, interferences will impact users using the conflicting networks.

**Proposed Solution:** In order to document known issues with hardware and software components, problem instances are created as members of the Problem class. An article describing a problem consists of a detailed description of the problem and a semantic description of the configurations which are affected by the problem. This allows for the creation of a list of all affected computers. If, for example, a problem exists which affects a certain type of computer model computer_model_1 in conjunction with a certain operating system operating_system_1 and a certain application application_1, the statement in the problem article look as follows:

```
{{#ask: 
[[Category:Computer]]
  [[has Model Type::computer_model_1]]
  [[has Installed::operating_system_1]]
  [[has Installed::application_1]]

[[has Solution::solution_1]]
```

This query returns a list of all computers of the specified model type with the mentioned operating system and application installed. In the last line, a link to the solution of the problem is given, which could include, for example, the recommendation to upgrade to a newer version of application_1. A computer, which would be returned by the query in the problem article would contain the following statements:

```
[[has Model Type::computer_model_1]]
[[has Installed::operating_system_1]]
[[has Installed::application_1]]
[[Category:Notebook]]
```

As can be seen, the described computer is part of the Notebook class, which itself is part of the Computer class. By using inferencing mechanisms, instances of subclasses of a queries class are returned. By using an external script, owners of computers are notified of problems on their computers and receive a link to the solution of the problem.
4 Related Work

There exist several implementations of commercial (e.g., Peregrine) as well as open source (e.g., i-doit\(^4\), OTRS::ITSM\(^5\)) solutions for managing CMDBs. However, these solutions lack features which support the collaborative editing of configuration information in agile environments. The specialized CMDB solutions known to the authors impose a rigid structure which lack the flexibility provided by the semi-structured semantic wiki approach. Implementing a CMDB and storing semi-structured information about configuration items in a semantic wiki, as presented in this paper, is a new approach which, in the authors’ view, simplifies the population of CMDBs and the maintenance of configuration information in agile environments.

5 Conclusions and Future Work

The present paper described the IT service management (ITSM) wiki, which supports a distributed, collaborative approach to configuration management in agile environments. Built on top of MediaWiki, Semantic MediaWiki and SMW+, the extension adds ITSM specific functions to the wiki. It was shown that it is feasible to use semantic wikis as a platform to implement a configuration management database. By making use of relations, attributes and categories, all configuration items as well as their relations to each other can be documented in a satisfying manner. Furthermore, by providing incentives to users of the documented systems, it is expected that documentation will be kept up to date by the users of the systems. Tab. 1 compares the features of specialized CMDB software, a wiki without semantic extensions and a semantic wiki. The table shows, that while CMDB software is strong in structural aspects, it lacks flexibility and collaborative features. Wikis are strong in collaborative aspects and the ability to describe information informally. Semantic wikis provide the strengths of specialized CMDB software and wikis by adding structure to wikis without sacrificing collaborative aspects and free text. Currently, there exists a testing implementation with about fifty configuration items. In order to fully leverage its potential, the ITSM wiki has to motivate employees to participate. While there exist traditional incentives for employees for getting work done in a corporate context (e.g., pay, promotion), [Andersen 05] suggest using normative incentives (e.g., praise, citations, peer respect, influence) for successful corporate wikis. Research on incentives in Wikipedia has been done by [Rafaeli and Ariel 08, Forte and Bruckman 05]. Patterns for the successful adaption of wikis can be found in [Mader (07)].

\(^4\) http://www.i-doit.org
In the future, the authors will test the proposed solution in a live environment with about two hundred users. It will be evaluated which of the incentives found in the literature can be realistically applied in our productive environment. Furthermore, a component for automatically populating and updating the wiki with configuration data will be implemented. In addition, a services and hosts monitoring component will be added, which allows the generation of configuration files for systems monitoring software from information expressed in semantic relations within the wiki. While the current implementation mainly exploits the collaborative aspects of wikis, future plans include making use of formal knowledge expressed in the wiki for improving conflict detection and recommending software to users. Furthermore, argumentation ontologies [Vrandečić et al. 05, Dellschaft et al. 08] will be evaluated as a tool for building consensus on configuration changes. In addition, an automatic check for conflicting changes, based on semantic descriptions, will be implemented.

Acknowledgements

We thank the anonymous reviewers for a number of useful comments. This work has partly been funded by the German Federal State of Baden-Württemberg and partly by the German Federal Ministry of Economics and Technology within project THESEUS.

References


