

# Automatic Population and Updating of a Semantic Wiki-based Configuration Management Database

Frank Kleiner, Andreas Abecker, Ning Liu

FZI Forschungszentrum Informatik  
an der Universität Karlsruhe (TH)  
{kleiner, abecker, ningliu}@fzi.de

**Abstract:** This paper describes our work on designing and implementing a component for automatically integrating and updating information about configuration items into a Semantic Wiki-based configuration management database. The presented solution uses technology for information gathering which is built-in or available for most current mainstream operating systems. By using Semantic Wiki technology, e.g., semantic queries and inference, the handling of configuration-management information is simplified and more powerful analyses are possible.

## 1 Introduction

IT landscapes which are becoming more and more complex as well as the need for faster and more flexible service delivery and the need for providing constantly available services, constitute an increasing challenge for the *IT service management* (ITSM) discipline. The work presented in this paper illustrates a particular aspect of our broader approach to collaborative, semantics-enabled ITSM support which is elaborated more comprehensively in [KA09] and [KAB09]. The work presented here allows for automatic population and updating of configuration management data stored in a Semantic Wiki by using standard-based mechanisms available for current mainstream operating systems. In addition to this automatically gathered information, Wiki articles representing computers are editable by members of the IT department as well as technologically knowledgeable users. This enables administrators, for example, to document the history of a computer (e.g., problems, support calls) next to the automatically gathered and updated information described in detail in this paper. In this introductory section, we give an overview of ITSM, Wikis, and Semantic Wikis, and sketch our overall approach.

### 1.1 IT Service Management

*IT service management* (ITSM) is the discipline which deals with all IT processes relevant for the successful achievement of a company's business goals. In contrast to IT management disciplines of the past, which dealt mostly with *technical* aspects of IT, 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 [CJ07, Köh07]. ITIL is a collection of best practices originally developed for the British government. ITIL defines processes, gives organizational guidelines, and describes tools for running IT infrastructures. Within ITIL, *configuration management* [LM07] is the process which deals with describing all entities used to provide IT services, as well as with describing the relationships between these entities. Entities relevant for configuration management are referred to as *configuration items* (CIs). Descriptions of configuration items and relationships between them are stored in the *configuration management database* (CMDB) which is the logical abstraction of all physical databases that contain information relevant for the configuration management in an organization [Köh07]. While there exists a variety of tools for automatically populating and updating CMDBs, not all information can be determined and put in context automatically.

## 1.2 Wikis and Semantic Wikis

Wikis are a special form of Web sites in which one may not only read published content, but can also add new and change existing information. While the Wikipedia<sup>1</sup> encyclopedia is the most prominent example, Wikis are often used as internal collaboration tools in companies or projects for facilitating knowledge management between coworkers. *Semantic Wikis* add semantic features to Wiki software. By adding explicit semantic statements [SS03, HKRS08], Wiki articles and their content as well as the interrelationships between articles and the entities described or represented by them, can be better processed by computers. There exist several different implementations of Semantic Wikis, each of which has its own characteristics and assumptions about how to make the best use of semantic technologies (see, for instance, [SBBK09, OVBD06]). When looking at the different implementations which merge Semantic technologies with Wiki approaches, it can be seen that Semantic Wikis can be divided into text-oriented Semantic Wikis and data-oriented Semantic Wikis. While the former put a high emphasis on text with added functionality for using semantics (e.g., Semantic MediaWiki<sup>2</sup> [VKV<sup>+</sup>06, VK09]) the latter add collaborative features to knowledge engineering systems (e.g., [ADR06]). While the use of data-oriented Semantic Wikis would be beneficial when only managing highly structured information about computer configurations, the component described in this paper is part of a wider approach which aims at providing a platform for managing and documenting all aspects of an IT infrastructure. This documentation of components, best practices and how-tos requires the use of a text-oriented Wiki. The component presented in this paper uses the freely available MediaWiki [Bar08] and the MediaWiki extensions Semantic MediaWiki (SMW) and SMW+<sup>3</sup> as the platform for the ITSM Wiki.

---

<sup>1</sup><http://www.wikipedia.org/>

<sup>2</sup><http://www.semantic-mediawiki.org/>

<sup>3</sup><http://wiki.ontoprise.com/>

### 1.3 Approach

The approach presented in this paper uses the *Windows Management Instrumentation* (WMI) feature of the Microsoft Windows operating systems for populating and keeping up-to-date a Semantic Wiki-based CMDB<sup>4</sup>. The use of a Semantic Wiki as the platform for retaining configuration data leads to a number of benefits, as described in [KA09], including collaborative aspects (from which agile environments with technologically knowledgeable users benefit most) as well as powerful processing mechanisms based on the formal meaning associated to links between configuration items and information stored in Wiki articles. By transforming information gathered via WMI, the component presented in this paper creates relations and attributes in the Semantic Wiki which represent the configuration management data. The use of the SMW query mechanism allows to generate tables with information gathered from multiple Wiki articles. The use of inferencing enables querying for information which is only implicitly stated in the Wiki.

## 2 Problem: Populating and Keeping Up-to-Date a CMDB

When looking at the configuration management database (CMDB) of a company, one can see a large quantity of information about hardware and software components. Initially populating the CMDB with all relevant information (e.g., CPU type and speed, or hard disk drive type and capacity) can be a cumbersome task. Additionally, maintaining and updating information can be even harder, even when only looking at the relatively static data. Examples of changed CMDB information can be the replacement of a hard disk by one with a greater capacity or an upgrade of the operating system. When taking into account dynamic data (e.g., the current amount of free space on a hard disk or the time of the last reboot of a computer), it is obvious that manual updates of this kind of information are not possible. The following subsections describe the problems with initially populating a CMDB, updating the more static information and updating the highly dynamic information in detail. Furthermore, the requirements for querying CMDB data are outlined.

### 2.1 Initial Population with Static Data

The initial population of a CMDB consists of gathering detailed information about all hardware and software components of an organization. Manually, this is not realistically feasible for organizations with more than about 10 computers. The number of errors to be expected with manual data collection is another factor which speaks against it. Automatic population mechanisms can be realized using mechanisms already present in the operating

---

<sup>4</sup>The use of WMI limits the hosts from which information can be gathered, to computers running Microsoft Windows. However, WMI is based on the *Web-Based Enterprise Management* (WBEM) standard which supports also Linux and Solaris operating systems. While the implementation of mechanisms for gathering information from Linux- and Solaris-based hosts is beyond the scope of this paper, the mechanisms could easily be implemented.

system or by the use of agents installed for this purpose. When gathering data, hardware components (e.g., computers and network equipment) are automatically scanned and relevant information is stored in the CMDB. Information gathered in the initial population includes the host name of a computer, the type of the hardware (e.g., notebook, desktop, server), the IP and MAC addresses, information about hardware components (e.g., CPU speed, number of CPUs, hard disk capacity, supported networking protocols,...), and information about installed software.

## **2.2 Updating Static Data**

Static data is data which rarely changes, e.g., the number of CPUs installed in a system or the capacity of the hard drive. While changes are relatively uncommon, they can occur, e.g., when a hard disk is replaced by one with higher capacity or a network address changes due to the replacement of a network-interface card or the placement in a different network segment. While these kinds of changes are relatively rare, they must be addressed by a system which deals with keeping a CMDB up-to-date. For keeping the CMDB up-to-date, a mechanism must be implemented which periodically checks known components for changes and also checks the network for components which are not yet present in the CMDB. When detecting a changed or new component, the CMDB has to be updated. In order to detect rogue or unwanted changes, the person responsible for the component (usually a system administrator) has to be notified. Furthermore, a history of changes has to be kept in order to provide accountability and to help in tracking down potential problems caused by the replacement of a component.

## **2.3 Updating Dynamic Data**

Dynamic data is defined as data which changes frequently and where changes usually do not have to be brought to the attention of the person responsible for the system. An example for dynamic data is the amount of free space on a hard disk drive. While this data changes frequently, it is fine to do so in most cases as long as there is enough space left on the disk. If the amount of free space drops below a certain threshold (usually around 15-20 percent), it makes sense to have a notification sent to the person responsible for the system. It can then be decided what measures to apply in order to mitigate the problem (e.g., delete unnecessary files, or replace the disk by one with higher capacity).

## **2.4 Flexible Querying**

Information stored in a configuration management database comprises a vast amount of data which has to be searchable and which can be grouped dynamically. The system has to be able to accept queries and generate information based on the query, e.g., in the form

of a table. A table containing all hardware of a certain type (e.g., all notebook computers), is a simple example for a dynamically generated table. Creating a table which contains only notebook computers with a certain processor and listing the operating system on that computers as well as the amount of RAM installed, is an example for a more complex query.

### **3 Accessing Systems Management Information**

Systems management deals with centralized management of a company's IT infrastructure. There exist several tools, protocols and standards for managing systems. The following subsections describe the standards used for automatically populating the Semantic Wiki-based CMDB.

#### **3.1 The Common Information Model (CIM)**

The Common Information Model (CIM) is a standard which is developed by the Distributed Management Task Force (DMTF)<sup>5</sup> as part of the Web-Based Enterprise Management (WBEM)<sup>6</sup> set of standards. It is used for modeling different aspects of an enterprise. CIM offers a "common definition of management information for systems, networks, applications and services, and allows for vendor extensions" [DMT09]. The common definitions help vendors of hardware and software components to exchange management information. CMI class instances represent actual hardware or software components. By querying class instances, information about real components can be accessed. Aspects of components can be modified by modifying the according class instance representing the component. CIM is language-independent, extensible and object-oriented [Mic09a].

#### **3.2 Windows Management Instrumentation (WMI)**

Windows Management Instrumentation (WMI) is the implementation of the Web-based Enterprise Management (WBEM) solution used in Microsoft Windows operating systems. WMI is based on the Common Information Model (CIM) standard (see section 3.1) which is developed by the Distributed Management Task Force (DMTF). WMI is used for implementing access to all kinds of manageable components (e.g., devices, networks and applications). By using the Distributed Component Object Model (DCOM) [Mic97] or the SOAP-based Windows Remote Management (WinRM) protocol [Mic09b], WMI allows remote access to systems management information. There exists support for most mainstream programming and scripting languages for accessing WMI information [Mic09a]. WMI implements a uniform interface for accessing management data, both,

---

<sup>5</sup><http://www.dmtf.org/>

<sup>6</sup><http://www.dmtf.org/standards/wbem>

remotely and locally. The uniform interface abstracts from the application programming interfaces (APIs) of the operating system. This enables applications and scripts to gather management data without having to have knowledge of the operating system API. Furthermore, functions can be accessed that otherwise would be accessed via APIs which are not designed for remote connections. Information which is made available through WMI, is used by accessing WMI classes. For making available information, a WMI provider has to be implemented. Figure 1 gives an overview over the parts of the architecture of WMI which are relevant for our work. WMI providers and managed objects form the bottom layer of WMI. This layer consists of the managed entities within the Microsoft Windows operating system and the CIMv2 WMI provider which gives access to the managed entities. All access to WMI is handled through the WMI infrastructure layer which includes the WMI core and the WMI repository. WMI consumers form the top layer which consists of the WMI COM API, the WMI Scripting API and the WMI scripts that access WMI through the Scripting API [Mic09a, Jon07].

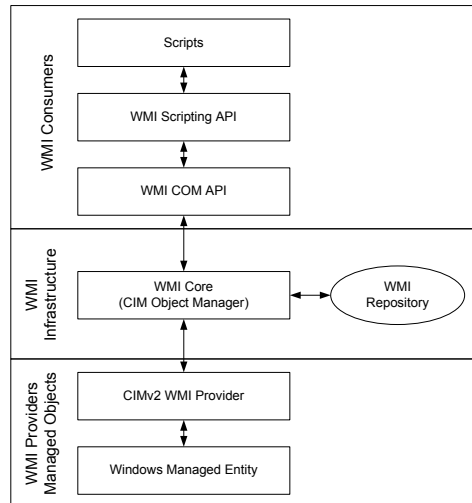


Figure 1: WMI Architecture (Picture adapted from [Mic09a])

Hardware components, e.g., hard disks, processors, and network adapters as well as the operating system, processes, and services are referred to as **managed objects**. Managed objects can be monitored by WMI providers. **WMI providers** implement an intermediate layer between WMI and managed objects. They are used for reading information from managed objects and adding the information to WMI and for handling messages from WMI for managed objects. WMI providers are implemented as Dynamic Link Libraries (DLLs) and a Managed Object Format (MOF) file that is used for defining classes for which the provider is responsible. **WMI classes** are defined in MOF files. Classes provide methods and properties to WMI consumers. The **WMI infrastructure** is the Microsoft Windows component responsible for providing WMI functionality. It consists of the WMI service and the WMI repository. The repository contains WMI-related static

data that is grouped by the use of namespaces. **WMI namespaces** are used for logically grouping different classes. Namespaces which are provided by Microsoft Windows are `root\default`, `root\cimv2`, and `root\subscription`. Additional namespaces can be created by providers implementing access to managed objects of additionally installed products (e.g., Microsoft Exchange or Microsoft SQL Server). Applications and scripts which interact with the WMI infrastructure are referred to as **WMI consumers**. WMI consumers are used to run queries, enumerate data, run methods offered by providers, or subscribe to events. Consumers can only access information of a managed object which is made available by the responsible provider. **The WMI Query Language (WQL)** is a subset of the SQL query language. It is used for retrieving class instances and data from WMI [Mic09a].

## 4 Architecture and Implementation

The architecture of our extension is shown in figure 2. Built on top of MediaWiki, Semantic MediaWiki and SMW+, the ITSM Wiki consists of a number of extensions which interact in order to provide an environment for collaboratively managing IT services. The component presented in this paper is the WMI Query Component, which is highlighted in grey in figure 2. The query component is responsible for populating the Wiki-based CMDB with information about new configuration items as well as to update changed configuration information. When encountering updated information, notifications can be generated. The query component plays the role of the WMI consumer when interacting with WMI.

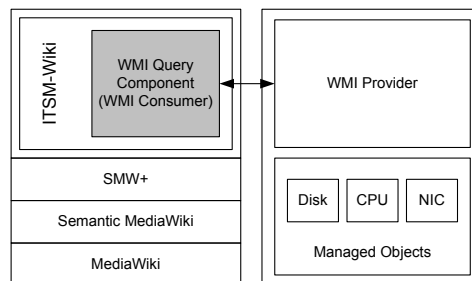


Figure 2: Architecture

Our solution is implemented in PHP<sup>7</sup> which allows access to the WMI infrastructure. The update component accesses WMI classes of remote computers and reads information contained in WMI properties. Reading of static information is usually done once a day, in order to keep a balance between causing network and CPU load on network switches and hosts, and keeping the ontology-based CMDB up-to-date. Because computers are not available for accessing WMI data at all times (e.g., due to switched off desktop computers

<sup>7</sup><http://www.php.net/>

or notebooks which are away), information is only updated if the WMI data can be read. If a computer is unavailable, the configuration information is kept in the state of the last successful read. In order to prevent the retaining of computers which are decommissioned, a time stamp is saved with each successful WMI data read. This allows to generate a Wiki article which includes all computers from which data was not read for a certain amount of time (e.g., two months).

## 5 The Configuration Item Ontology

Figure 3 shows a part of the configuration item ontology which forms the data model underlying this work. The ontology is modeled in KAON<sup>8</sup> ontology management toolsuite. Classes are displayed as rectangles with lines connecting super-classes and sub-classes. Properties (relations and attributes) are shown as hexagons, with arrows indicating the direction of the property. The **Hardware** class is one of the central classes of the ontology. It is the super-class of all hardware components. The attributes **Serial Number**, **Model Number**, **Manufacturer**, and **Model** are defined for all hardware components. The class **Computer** is a sub-class of **Hardware**. It represents all computer systems and includes the sub-classes **Desktop**, **Notebook**, and **Server**. Processors are described in the **CPU** class. The class **Peripheral** is another sub-class of the **Hardware** class. It comprises all peripheral components, e.g., a computer mouse or a keyboard, which are represented by their respective classes. Peripherals are connected to computers which is expressed with the **Peripheral is connected to Computer** relation. Storage devices are another example for hardware. This type of hardware is specified in the **Storage Device** class. The sub-class of **Storage Device** is the class **Hard Disk**, which has a capacity, which is expressed in the attribute **Capacity**. Storage devices are connected to computers via the **Storage is used in** relation. Extension cards are modeled in the **Extension Card** class. Examples for extension cards are graphics adapters, network adapters (NICs), and sound adapters, which are modeled in the classes **Graphics Adapter**, **Network Adapter**, and **Sound Adapter**. Network adapters are usually assigned a hardware address (MAC address) and an IP address. Because there exist scenarios where there are network adapters with multiple IP addresses, and because IP/MAC address bindings can change, e.g., when replacing a network adapter, a helper concept is included in the ontology. The helper concept, named **Network Configuration** allows the use of n-ary relations. A network configuration instance usually includes the MAC address of the network adapter and the IP address which is assigned to a computer. For special cases, the n-ary relations allow for more complex configurations, e.g., a network adapter with multiple IP addresses. The **Software** class is another class represented in the ontology. In the example, it contains the sub-classes **Application Software** and **Operating System** as well as **Service Release**. The relation **has installed Software** between the **Computer** class and the **Software** class states which software instances are installed on an instance of the computer class.

Up to now, we mainly modeled the class hierarchy, attributes, and relationships—no cardinality constraints, more complex integrity constraints, or any rule knowledge yet. Future

---

<sup>8</sup><http://kaon.semanticweb.org/>

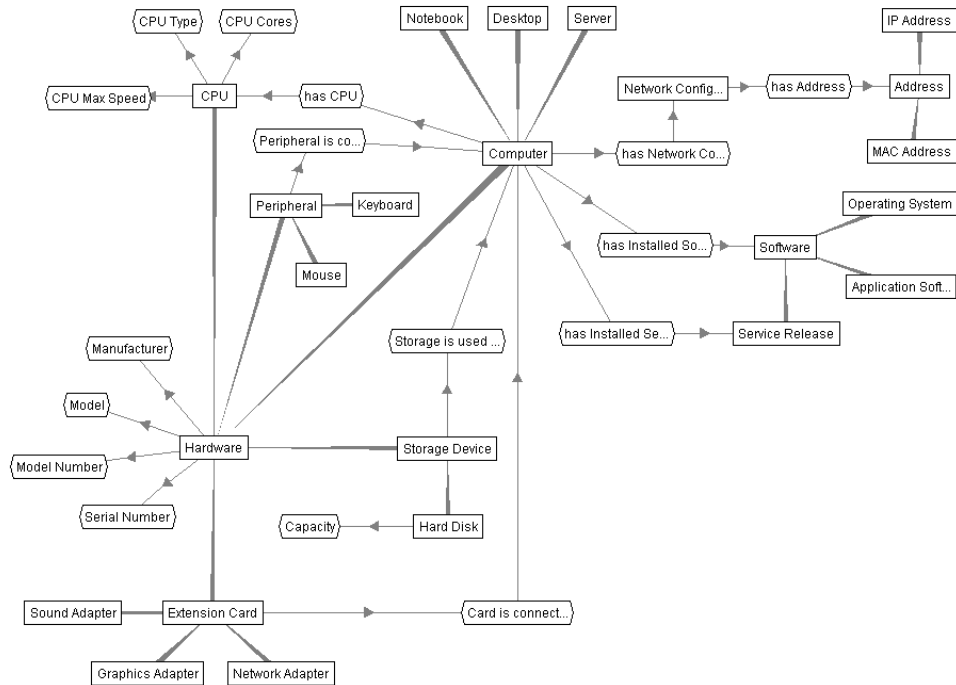


Figure 3: Excerpt of Configuration Item Ontology

usage and implementation of use cases will show how heavyweight the ontology must be for this application domain.

## 6 Application Example

Each configuration item is represented in a Wiki article. Figure 4 shows a screenshot of a concrete computer represented in the Wiki. There are several sections in the article, each of which deals with a different aspect of this computer. All entries shown in the Wiki article are automatically gathered by the component described above. The **hardware information section** includes information about the CPU(s) of a computer, e.g., the processor type, as well as model number and serial number of a computer. Information about the **operating system** includes the operating system name, the version of the operating system and the installation date. The **networking section** comprises the hardware and IP addresses of the various network interfaces. In the **storage section**, information about storage devices, e.g., hard disks is grouped. The **application section** displays information about all applications installed on the computer represented in this Wiki article. For lack of space, the storage and application sections are omitted in the screenshot. The Wiki text below gives

an impression of the representation of the hardware information section in SMW syntax:

```
* Model Number: [[Model Number::20078JG]]
* Serial Number: [[Serial Number::LX3TXXX]]
* CPU Type:
  [[has CPU::Intel(R) Core(TM)2 CPU T7600 @ 2.33GHz]]
```

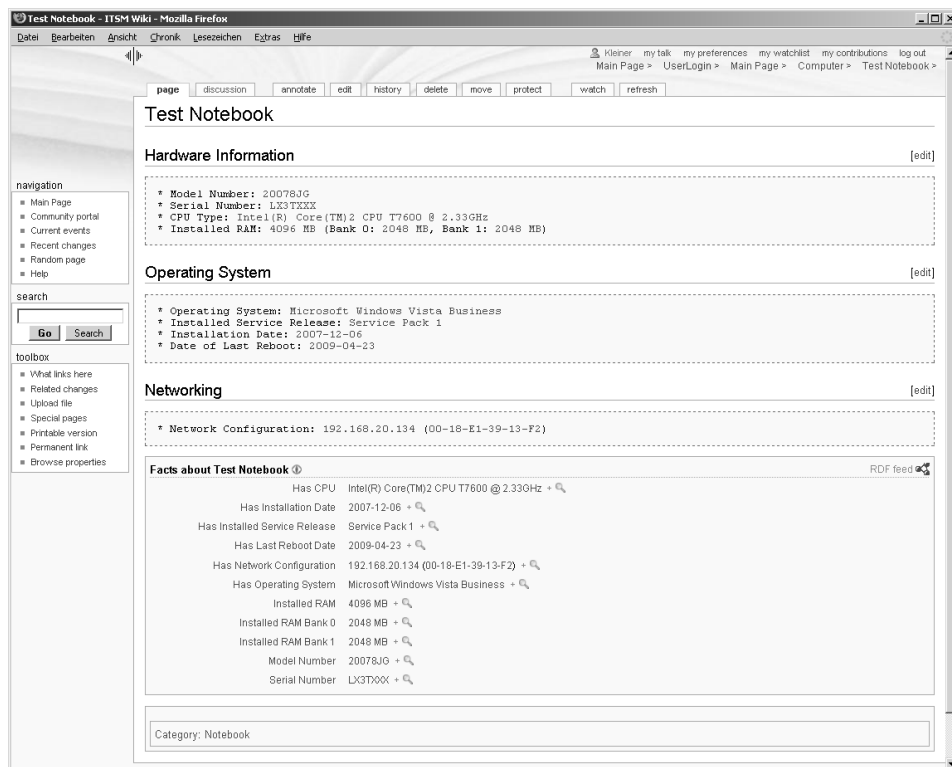


Figure 4: Screenshot of a Sample Notebook Represented by an ITSM Wiki page

## Use of Semantic Queries

The Semantic MediaWiki platform provides inline queries<sup>9</sup> for semantically querying information stored in the Wiki. Furthermore, a connector for the Jena<sup>10</sup> Triple Store can be added to the Wiki in order to enable more complex queries which exceed the capa-

<sup>9</sup>[http://semantic-mediawiki.org/wiki/Help:Inline\\_queries](http://semantic-mediawiki.org/wiki/Help:Inline_queries)

<sup>10</sup><http://jena.sourceforge.net/>

bilities of inline queries. Inline queries are initiated with the `#ask:` keyword. Because all configuration-item information gathered by the component described in this paper, is stored in semantic properties, inline queries can be used for dynamically creating surveys of information found in relations and attributes. An inline query creating a table which lists all notebook computers (`[[Category:Notebook]]`) with a specific CPU (`[[has CPU::Intel(R) Core(TM)2 CPU T7600 @ 2.33GHz]]`) and the Microsoft Vista operating system (`[[Operating System::Microsoft Windows Vista Business]]`) looks as follows:

```
{#{ask:
 [[Category:Notebook]]
 [[has CPU::Intel(R) Core(TM)2 CPU T7600 @ 2.33GHz]]
 [[has Operating System::
      Microsoft Windows Vista Business]]
 | ?has Installed Service Release
 | ?has Installation Date
 | sort=has Installation Date
 | order=descending}}
```

The generated table includes the name of the found Wiki article—which in this case is the computer name—the service-pack version of Microsoft Vista installed on this computer (`?has Installed Service Release`) and the installation date of the operating system (`?has Installation Date`). The table is sorted by installation date (`sort=has Installation Date`) with most recent installations shown first (`order=descending`).

In order to enable more complex queries, the Jena Triple Store Connector<sup>11</sup>, which is an extension to SMW+, is used. Complex queries are, for example, ones which include a combination of different subjects in a query. The Triple Store Connector connects an externally running Jena instance which can be queried by using the SPARQL query language<sup>12</sup>. Queries can be used both, from within the Wiki as well as from external programs, which can query for all semantic information stored in the Wiki. The possibility for using more complex queries comes at a price, however: While the use of inline queries by the use of the `#ask:` keyword is manageable by regular users after some basic training, the use of SPARQL queries is far more complex.

The use of Semantic Web technologies for managing configuration information provides several benefits when using queries: The use of **inferencing**<sup>13</sup> enables to infer new information from existing information. For example, an inline query can generate a table including all computers and all versions of Microsoft Windows instead of just notebooks and a particular version of Windows. Because there exists a hierarchy of hardware components (e.g., the notebook category is a sub-category of the computer category), a query for computers in the `Computer` category, which represents all computers, also returns all computers found in the `Notebook`, `Server` and `Desktop` categories. The same is

<sup>11</sup>[http://semanticweb.org/wiki/Halo\\_Extension](http://semanticweb.org/wiki/Halo_Extension)

<sup>12</sup><http://www.w3.org/TR/rdf-sparql-query/>

<sup>13</sup><http://semantic-mediawiki.org/wiki/Help:Inferencing>

true for queries for software components, e.g., the operating system installed on a computer. Because of the hierarchical structure, a query for all computers running Microsoft Windows returns all computers running, amongst others, Windows Vista and Windows XP. This is accomplished by placing software versions into a category hierarchy (e.g., the categories `Windows Vista` and `Windows XP` as sub-categories of `Microsoft Windows`). Such an exploitation of transitive closures or even more complex (e.g., rule-based) inference knowledge is much more flexible and powerful than reporting mechanisms in state-of-the-art CMDB software. When using complex queries by using the SPARQL query language, even more powerful reporting mechanisms are available to advanced users.

Such flexible and powerful query mechanisms can find many useful applications in practice where new, very specific reports may become necessary frequently, maybe even as ad-hoc queries formulated by an experienced system administrator—which can not be hardcoded in a CMDB tool beforehand; example usage scenarios might be: a new virus has become known which infects only computers with specific configurations of operating system version, patches, and installed software; one user identifies a system bug and other systems with similar hardware-software configuration must be identified preventively; some RAM modules have become available because of a system reconfiguration, which might be reused in another computer provided all hardware could be identified which could be reconfigured with the newly available hardware; for strategic planning of hardware purchases, a list of systems with recurring bottleneck behavior wrt. important resources shall be compiled.

## 7 Related Work

There exist several systems management solutions, both freely and commercially available. A good example for a freely available software is SpiceWorks<sup>14</sup> which includes a component for managing inventory (e.g., hardware, software). Detailed hardware and software configuration information is extracted from computers via the network and can be displayed and searched. Hardware can be grouped by a matching criterion (e.g., the same model number of a computer) or by creating custom groups which use several criteria (e.g., all computers from a certain manufacturer running a certain operating system and having more than a specified amount of RAM) to decide if a computer is member of the group. A reporting mechanism allows for generating reports based on several criteria and outputs a custom table. While SpiceWorks allows for custom groups, these groups only allow one hierarchy layer, i.e., for example, the workstation group can consist of laptops and desktops, but laptops cannot consist of other nested classes like netbooks or tablet PCs. The SMW technology, in conjunction with the Jena Triple Store accessible via SPARQL queries, offers much more flexible and powerful mechanisms for querying data. Our overall approach which is characterized by a Semantic Wiki as the central information hub between structured, automatically collected data and unstructured, manually edited information about user's personal notes, experiences, error reports, etc, maybe even extended

---

<sup>14</sup><http://www.spiceworks.com/>

by external information streams (product information, bug reports,...)—all that integrated through the semantic backbone—is far ahead of state-of-practice CMDB software.

Surprisingly, even in the area of Semantic Web research, not much, recent, serious work can be found about ITSM ontologies or semantic configuration item management. [BP07] discuss the problem of differing implementations of the CIM standard in heterogeneous environments. The use of ontologies for formally representing IT resources is suggested. A longer thread of work, represented, e.g., by [dVVB04, GVdV<sup>+</sup>06, SdVT<sup>+</sup>09], uses OWL for defining network and system management information, for several purposes such as configuration management, network management policies, and network traffic control. The authors have comprehensive models of hardware aspects, but very few regarding the software, service, or organizational aspects of ITSM. On the other hand, [BDL<sup>+</sup>07] presents a very comprehensive, top-down integration model of all ITSM related key domains—organization, process, tools, technology—and their interrelationships, but it remains on a very abstract, non-technical level. In our future work, based on first practical experiences made with the already built ontology sketched above, we will aim at the creation of a reference ontology for configuration items, widely usable, driven by practical requirements, yet carefully modeled and quality-assured.

## 8 Summary, Conclusions and Future Work

An extension of the Semantic MediaWiki-based ITSM Wiki was presented which allows the WMI-based, automatic population of the ITSM Wiki with information gathered from computers running Microsoft Windows. This information is stored in Semantic Wiki articles and enriched with semantic information. The addition of semantic information allows to use special queries and inferencing which would not be possible otherwise. After giving an introduction to IT service management, Wikis and Semantic Wikis, it was shown how hardware and software information can be obtained via the WMI infrastructure. Then the architecture and implementation of our component were presented, followed by a sketch of the Configuration Item ontology that forms the data model underlying our solution. The subsequent sections presented an application use case and a short discussion of related work.

It should be noted critically that retaining information about computers raises security and privacy concerns. Configuration data stored in a Wiki is a potential source of information which might be used for planning an attack on a computer. Furthermore, information about configurations can contain elements which allow to draw conclusions about personal data or behavior of a person (e.g., what software is installed on a computer, or how much disk space is used). In order to ensure security and privacy, access control mechanisms have to be implemented that restrict access to Wiki articles. It must be ensured that sensitive data is excluded from results returned to queries. Until the completion of a more sophisticated security model, the built-in access control mechanisms of MediaWiki will be used to restrict sensitive articles.

Future work will include support for additional WMI classes for gathering data from

installed applications (e.g., Microsoft Exchange, Microsoft SQL Server, and Microsoft Office). Moreover, our software component will be extended by adding support for accessing information which is made available through the Simple Network Management Protocol (SNMP) [CFSD90, CMPS02] that will enable to include static and dynamic information about devices, e.g., printers and network switches. Information gathered from these devices includes network configurations (MAC and IP addresses), hardware information (manufacturer, serial numbers) and usage information (paper and toner available in a printer, or network load on switch ports). Further research will also be done on how more expressive representation languages (e.g., OWL) can be used and which benefits can be gained from them. The presented solution will be evaluated in a productive environment consisting of 500 computers in the third quarter of 2009.

## Acknowledgements

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

## References

- [ADR06] S. Auer, S. Dietzold, and T. Riechert. OntoWiki - A Tool for Social, Semantic Collaboration. In I.F. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. Aroyo, editors, *International Semantic Web Conference*, volume 4273 of *Lecture Notes in Computer Science*, pages 736–749. Springer, 2006.
- [Bar08] D.J. Barrett. *MediaWiki (Wikipedia and Beyond)*. O'Reilly Media, 1st edition, October 2008.
- [BDL<sup>+</sup>07] J. Black, C. Draper, T. Lococo, F. Matar, and C. Ward. An integration model for organizing IT service management. *IBM Systems Journal*, 46(3):405–422, 2007.
- [BP07] K. Breitman and M. Perazolo. Using Formal Ontology Representation and Alignment Strategies to Enhance Resource Integration in Multi Vendor Autonomic Environments. *EASE '07: Proceedings of the Fourth IEEE International Workshop on Engineering of Autonomic and Autonomous Systems*, pages 117–126, 2007.
- [CFSD90] J.D. Case, M. Fedor, M.L. Schoffstall, and J. Davin. Simple Network Management Protocol (SNMP). RFC 1157 (Historic), May 1990.
- [CJ07] B. Clacy and B. Jennings. Service Management: Driving the Future of IT. *Computer*, 5:98–100, 2007.
- [CMPS02] J. Case, R. Mundy, D. Partain, and B. Stewart. Introduction and Applicability Statements for Internet-Standard Management Framework. RFC 3410 (Informational), December 2002.
- [DMT09] DMTF. Common Information Model (CIM) Standards. Web: <http://www.dmtf.org/standards/cim/>, March 2009.

- [dVVB04] J.E.L. de Vergara, V.A. Villagra, and J. Berrocal. Applying the Web ontology language to management information definitions. *Communications Magazine, IEEE*, 42(7):68–74, July 2004.
- [GVdV<sup>+</sup>06] A. Guerrero, V.A. Villagr a, J.E. L opez de Vergara, A. S anchez-Maci an, and J. Berrocal. Ontology-Based Policy Refinement Using SWRL Rules for Management Information Definitions in OWL. In R. State, S. van der Meer, D. O’Sullivan, and T. Pfeifer, editors, *DSOM*, volume 4269 of *Lecture Notes in Computer Science*, pages 227–232. Springer, 2006.
- [HKRS08] P. Hitzler, M. Kr otzsch, S. Rudolph, and Y. Sure. *Semantic Web – Grundlagen*. Springer, Berlin, 2008. In German.
- [Jon07] D. Jones. *VBScript, WMI, and ADSI Unleashed: Using VBScript, WMI, and ADSI to Automate Windows Administration (Unleashed)*. Addison-Wesley Longman, Amsterdam, 2nd edition, June 2007.
- [KA09] F. Kleiner and A. Abecker. Towards a Collaborative Semantic Wiki-based Approach to IT Service Management. 2009. Submitted to I-SEMANTICS 2009.
- [KAB09] F. Kleiner, A. Abecker, and S.F. Brinkmann. WiSyMon - Managing Systems Monitoring Information in Semantic Wikis. 2009. Submitted to WikiSym 2009.
- [K oh07] P.T. K ohler. *ITIL: Das IT-Servicemanagement Framework (Xpert.Press)*. Springer, Berlin, 2nd edition, 2007. In German.
- [LM07] S. Lacy and I. Macfarlane. *Service Transition (ITIL)*. The Stationery Office Ltd, May 2007.
- [Mic97] Microsoft MSDN Library. DCOM Architecture. Web: [http://msdn.microsoft.com/de-de/library/ms809311\(en-us\).aspx](http://msdn.microsoft.com/de-de/library/ms809311(en-us).aspx), July 1997.
- [Mic09a] Microsoft MSDN Library. Windows Management Instrumentation. Web: [http://msdn.microsoft.com/en-us/library/aa394582\(vs.85\).aspx](http://msdn.microsoft.com/en-us/library/aa394582(vs.85).aspx), April 2009.
- [Mic09b] Microsoft MSDN Library. Windows Remote Management. Web: [http://msdn.microsoft.com/en-us/library/aa384426\(vs.85\).aspx](http://msdn.microsoft.com/en-us/library/aa384426(vs.85).aspx), February 2009.
- [OVBD06] E. Oren, M. V olkel, J.G. Breslin, and S. Decker. Semantic Wikis for Personal Knowledge Management. In S. Bressan, J. K ung, and R. Wagner, editors, *DEXA*, volume 4080 of *Lecture Notes in Computer Science*, pages 509–518. Springer, 2006.
- [SBBK09] S. Schaffert, F. Bry, J. Baumeister, and M. Kiesel. Semantische Wikis. In A. Blumauer and T. Pellegrini, editors, *Social Semantic Web*, X.media.press, pages 245–258. Springer, 2009. In German.
- [SdVT<sup>+</sup>09] A. Salvador, J.E. L opez de Vergara, G. Tropea, N.B. Melazzi,  . Ferreiro, and A. Guerrero. Ontology Design and Implementation for IP Networks Monitoring. In *1st Int. Workshop on Web & Semantic Technology (WeST-2009)*, Wrexham, North East Wales, UK, 2009.
- [SS03] S. Staab and R. Studer. *Handbook on Ontologies. (International Handbooks on Information Systems)*. Springer, Berlin, 1st edition, 2003.
- [VK09] D. Vrande i c and M. Kr otzsch. Semantic MediaWiki. In J. Davies et al., editor, *Semantic Knowledge Management*, pages 171–179. Springer-Verlag, 2009.
- [VKV<sup>+</sup>06] M. V olkel, M. Kr otzsch, D. Vrande i c, H. Haller, and R. Studer. Semantic Wikipedia. In *WWW ’06*, pages 585–594. ACM Press, 2006.