Requirements and Recommendations for the Realization of a Configuration Management Database

Thomas Schaaf[1], Boran Gögetap[2]

[1] MNM Team, Ludwig Maximilians University, Munich, Germany
Thomas.Schaaf@mnm-team.org
[2] Munich Institute for IT Service Management (mITSM), Munich, Germany
Boran.Goegetap@mitsm.de

Abstract. In IT Service Management – as it is proposed by the IT Infrastructure Library (ITIL) – Configuration Management is considered as a vital discipline for effective support of IT service operation, well directed response to incidents, successful planning of changes and controlled roll out of software releases in an IT environment. The Configuration Management Database (CMDB) therefore has to serve as a comprehensive knowledge base containing information on IT services, hardware, software and much more.
The requirements on a CMDB are manifold. Today, various tools from different vendors exist, but none of them may claim for itself to fit every imaginable demand. That is why, in practice, individual CMDB solutions are developed.This paper presents a first move in giving generic, conceptual recommendations for implementing or customizing a CMDB solution, based on the requirements that have to be fulfilled.

1 Introduction & Problem Statement

The *IT Infrastructure Library (ITIL)* is a collection of books in which best practices in IT Service Management are described. ITIL follows the principle of process-oriented (IT Service-) Management and groups management activities into defined management processes. Incident Management, Problem Management, Change Management and Release Management are four of the five operative ITIL Service Management processes described in [6]. The fifth process is the *Configuration Management* process that plays a decisive role in this context: Its main task is to provide information on the IT environment to the other processes. For example, if the same incident has been reported by a number of users (e.g. outage of service e-mail), the Incident Management process will be interested in information on which other users might be affected (impact) and what workaround could be enabled in order to reduce service downtime. The Problem Management process would want to analyze potential causes for the incident (e.g. router failure) and perform a Fault Tree Analysis (FTA) for that purpose. All this is only possible, if the *Configuration Management Database*

(CMDB) contains up-to-date and relevant information on the affected Configuration Items (CIs).
Problem Statement: In this paper we take two steps forward from these goals toward a solution ready for use in an operational management environment by addressing the following two basic questions: Step I – What are the requirements on a CMDB tool and its underlying information model? And Step II – How can these requirements be accomplished when implementing or choosing and customizing a CMDB tool? Our results help bridging the gap between *management requirements* and the *implementation concepts* needed in the context of a CMDB.

2 Solution Step I: Requirements Analysis

The goal of a CMDB is to store management-relevant information for every identified *Configuration Item (CI)*. A narrow definition for a CI – similar to the one given in [6] – is the following one: *Component of or associated with an infrastructure that contributes to the delivery of an IT service and that is (or is to be) under the control of Configuration Management.* A more universal definition will probably be used in the next release of ITIL (ITIL V.3, scheduled for this year) and calls *any item that is involved into the delivery of an IT service – including the IT service itself* a CI. Most practitioners and researchers already use the second, extended definition, and so do we as a starting point for our considerations.

In the following, we specify two sets of requirements for a CMDB (tool): The first set refers to the *underlying information model*. Accordant to [5] the information model contains rules for the description of the managed objects – in this case for the description of the CIs. The second set covers *functional requirements* on a CMDB tool and provides a basis for the Configuration Management functions and activities that need to be supported by the CMDB. In advance of the presentation of the respective requirements, we give a short outline of the methodology of deriving and developing those requirements.

2.1 Requirements on the Information Model

A CMDB information model will need to fulfill a number of requirements. We already addressed this issue earlier in [3]. The following list is a partial rework of the criteria presented in this article.
Methodology for developing information model requirements: *For this set of requirements, we particularly analyzed the* ***information structure*** *of each ITIL process ([6], [7]) that is basically reflected in the* ***process-specific artifacts****. In addition, we investigated the* ***information and data flows*** *between processes as well as the* ***general characteristics*** *of the ITIL framework concerning management-relevant information.*

IM1 **Adaptability of Model** All ITSM processes are subject to a continuous improvement cycle. Consequently, the CMDB must be capable of dealing

with changing requirements, especially regarding scope, nature and level of detail of the documented information, resulting in dynamic adaptability of the information model.

IM2 **Alignment to ITSM information needs** The information model for a CMDB should address information requirements of the ITSM processes and consequently either include or reference models of all relevant entities (e.g. Incident Records, Change Requests (RfCs), etc.).

IM3 **Comprehensive view on infrastructure and component relations** The documentation of CI relationships (e.g. for performing an impact analysis) is maybe the single most essential concept in the CMDB context. Consequently, the information model should include basic relations between common CI types and support modeling multiple relationships between CIs.

IM4 **Support for life cycle status accounting** ITIL demands that the life cycle status of any CI is tracked and documented. This should be reflected in the information model. Also information pertaining to all life cycle phases should be accessible through a CMDB.

IM5 **Catalog of basic CI types** Provisioning of common CI types – preferably in the form of an extendable but ready-to-use data models – could significantly shorten the time-to-implementation for a CMDB.

2.2 Functional Requirements on a CMDB tool

The above requirements refer to the information model building the foundation for a CMDB. But some crucial requirements can not be covered by the information model. Those are the requirements we subsume as functional requirements on a CMDB tool.

Methodology for developing functional requirements: *In order to derive functional requirements, we analyzed the* ***tasks and activities*** *defined within the ITIL process work flows with a particular focus on the activities in the Configuration Management process as defined in [7].*

F1 **CI identification support** The identification of CIs belongs to the second activity in the ITIL Configuration Management process and requires adequate support by the CMDB tool, in particular with respect to naming conventions, data consistence and the recording of a configuration baseline. In large environments, the latter will require auto discovery mechanisms – either performed by the tool itself or imported from an external source.

F2 **Visualization of (parts of) the CMDB** An adequate visualization of the stored information (including automated and intuitive data partition and graph organization) is essential in order to support several activities and tasks in the Configuration Management Process like reviews and audits. For example, in a visualized extract of the CMDB, "islands" indicating missing relationships can be identified.

F3 **Component Failure Impact Analysis (CFIA)** Another mandatory CMDB tool function is the CFIA that helps a user to quickly find for one CI any supporting CIs (top-down) or any supported CIs (bottom-up).

F4 **Plausibility Checks and Audit Support** Verification and Audit is the fifth activity in the Configuration Management process defined by ITIL and should be supported by the CMDB tool. We suggest three types of audits that should be supported: A content-related audit aims at the discovery of unauthorized (unconfirmed) CIs and unknown instances (e.g. "unknown location"). In a structural audit, isolated CIs (cf. F2) or shortfalls of minimum cardinalities are detected. As a third type of audit, technical consistency checks are intended to find duplicate CIs or invalid relationships.

F5 **Integration with external databases and systems management data stores** Information of relevance might be managed and stored outside the IT organization – either in enterprise databases or in external CMDBs. A CMDB tool should therefore ease reconciliation of data stored in the CMDB with that of other existing data stores and management systems.

3 Solution Step II: Realization Recommendations

In step two of our solution, the above requirements analysis is followed by generic implementation recommendations in order to provide guidance for the realization or customization of a CMDB (tool). The recommendations will basically be independent from a concrete programming paradigm or existing tool.

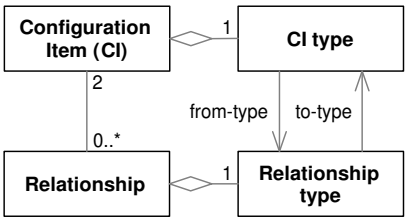

Fig. 1. Simple abstract CMDB model

The only assumption we make is the simple and abstract CMDB model depicted in Figure 1. Accordingly (and in compliance with [6] and IM3), the two main concepts provided by a CMDB are CI and Relationship, for each of which different types (classes in case of an object-oriented implementation) should be elaborated for the regarded use case. Different CI and Relationship types are not always compatible to each other (e.g. an "is located" relationship can be established between a CI of the type router and one of the type location, but not between two CIs of the type router).

Be aware that this simple abstract model does not say anything about implementation or details on the information structure of the modeled services, resources and process artifacts. But for the convenience of a common understanding of the basic terms and concepts in CMDB modeling, it provides a sensible foundation which is now followed by more concrete implementation and modeling recommendations. In addition, we consider the CMDB to be a *federated system*, accessed by *multiple users* and providing common basic *database capabilities* including queries and different logical views on the data.

R1 **Provide CI type revisions** CI types may require modifications while they are already in use. Within the continuous improvement cycle (cf. IM1), the Configuration Management-relevant attributes may change (e.g. adding new

attributes and/or replacing existing ones). In order to support dynamic CI types, numeric revisions can be used to easily differentiate between an older and the current type version. It is important that CI instances of older CI type revisions keep operating or can be migrated to the new type revision, if desired.

R2 **Provide (at least) two basic data confirmation concepts: register and submit** Differentiating between a preliminary registration and the final submission of a CI, Relationship, CI type or Relationship types allows the Configuration Manager to involve a multitude of persons (IT staff) into the process of filling the CMDB by at the same time retaining control of the progress by sparingly distributing final submission grants.

R3 **Support relationship cardinalities and cardinality compliance checks** Possible cardinalities of relationships are 1 to 1, 1 to n and m to n. Being able to assign a specified cardinality to a certain Relationship type can be one important means of ensuring a consistent Relationship modeling. Since Relationship types may exist in different revisions (cf. R1), a compliance check between Relationships and the allowed cardinality defined in the Relationship type should not be limited to the point in time of registering or submitting a Relationship.

R4 **Provide placeholders: Dummy, Joker and Default** One major specific of a CMDB is that its setup usually spans a long time period in which on the one hand CIs, Relationships and types of both are added, and on the other hand the underlying concrete model (in a relational approach: the schema; in an object-oriented approach: the classes and attributes) changes within the process of continuous improvement (cf. IM1). One problem in this context is that maybe not all information are available when they are needed. In order to face this problem, Dummies, Jokers and Defaults can be used as information placeholders.

R5 **Provide user-specific task lists** Effectiveness and efficiency of the Configuration Management process can be increased by clearly delineating responsibilities as well as correct prioritization of incidental tasks. In a multi-user CMDB solution, separate task lists for all users are a powerful means to support this goal.

R6 **Support visualization by CI type-specific view levels** The visualization of (extracts of) the CMDB can be facilitated by assigning view levels in terms of numeric values to the CI types. One example of an effective application of view levels is to assign a low view level to CI types near the resource level and a high view level to CI types near the business process level. This way, the visualization of hierarchies (vertical extracts) or specific layers (horizontal extracts) becomes possible, if the visualization engine is capable of dealing with this concept.

4 Current Status & Related Work

Current status and next steps: The above listed are only six out of a larger quantity of implementation recommendations for a CMDB (tool) that have been

developed in a joint effort of the mITSM and the MNM Team. Currently, a more comprehensive catalog of recommendations is being developed, while at the same time the existing recommendations are further elaborated.Thus, the considerations in this short paper reflect first steps and early results from our research in the broad area of automation and tool support in IT Service Management.

In order to evaluate the recommendations, we have developed a first prototype of a free CMDB tool that already puts the majority of concepts into action, while the still missing ones will be realized gradually in the future (available at [8]). In future projects, we will perform an evaluation using a real IT environment.

Related Work: In the field of CMDB design and tool support for IT Service Management accordant to ITIL, various white papers and studies (see e.g. [2], [4]) have been published. Recently, an approach toward a "Federated CMDB" has been published as a joint white paper of BMC, CA, Fujitsu, HP, IBM and Microsoft [1]. In addition, there are papers available from BMC, CA, HP, IBM and other vendors that describe specific (commercial) CMDB solutions.

Generally, the market for CMDB tools is growing rapidly, since ITIL increasingly gains attention especially in large and medium-size enterprises. Today, it is already hard to keep track of the variety of functionalities different tools provide. This makes it even more difficult to approach this topic in a way satisfactory to the entire researchers' and vendors' community. That is why we pursuit a structured, analytical and generic approach in CMDB design that does not claim for itself to cover any imaginable implementation detail, but is rather focused on requirements analysis-based realization recommendations.

References

1. BMC, CA, Fujitsu, HP, IBM, Microsoft. *The Federated CMDB Vision*, 2007.
2. BMC Software Inc. *What do you need from a Configuration Management Database (CMDB)?*, 2005.
3. M. Brenner, M. Sailer, T. Schaaf, and M. Garschhammer. CMDB – Yet Another MIB? On Reusing Management Model Concepts in ITIL Configuration Management. In *Proceedings of 17th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM 06)*, volume 2006, pages 269–280, Berlin, Germany, October 2006. Springer.
4. Ronni J Colville. *CMDB or Configuration Database – Know the Difference*. Gartner, March 2006. Research Note G00137125.
5. H.-G. Hegering, S. Abeck, and B. Neumair. *Integrated Management of Networked Systems – Concepts, Architectures and their Operational Application*. Morgan Kaufmann Publishers, ISBN 1-55860-571-1, 1999. 651 p.
6. Office of Government Commerce (OGC), editor. *Best Practice for Service Support*. IT Infrastructure Library (ITIL). The Stationary Office, Norwich, UK, 2000.
7. Office of Government Commerce (OGC), editor. *Best Practice for Service Delivery*. IT Infrastructure Library (ITIL). The Stationary Office, Norwich, UK, 2001.
8. Sourceforge project. BPMspace for ITIL, December 2006.