Logical Design for Configuration Management Based on ITIL

Conference Paper · June 2012

CITATIONS
0

READS
655

2 authors:

Hristo Kyurkchiev
Sofia University "St. Kliment Ohridski"
6 PUBLICATIONS **4** CITATIONS
SEE PROFILE

Kalinka Kaloyanova
Sofia University "St. Kliment Ohridski"
29 PUBLICATIONS **23** CITATIONS
SEE PROFILE

Logical Design for Configuration Management Based on ITIL

Hristo Kyurkchiev[1], Kalinka Kaloyanova[1]

[1]Faculty of Mathematics and Informatics, Sofia University, 5 James Bourchier blvd., 1164, Sofia, Bulgaria

hkyurkchiev@fmi.uni-sofia.bg, kkaloyanova@fmi.uni-sofia.bg

Abstract. The focus of this paper is the Configuration Management Database (CMDB) as seen by Information Technology Infrastructure Library (ITIL). It aims at providing a conceptual and logical model for this database. The core theoretical CMDB concepts are considered and also many practical problems are discovered. Having uncovered the main obstacles and problems that an implementer faces, valid solutions are provided. As a result a logical model of a CMDB is proposed. Its practical value for implementation has been tested with real life data and satisfactory results are reached.

Keywords: IT Services, ITIL, Configuration Management Database, Knowledge Management, Logical Database Model

Introduction

IT services are one of the sectors in the world's economy that practically grows by the minute. In this environment, a knowledge reference for managing IT services was created by the Office of Government Commerce (OGC) [10]. Ever since its first version the Information Technology Infrastructure Library or ITIL has been pushing the envelope to making better, more reliable IT services through the adoption of some basic guidelines and best practices. As of the third and last version of the library this is done through separating the service lifecycle into phases and the phases themselves into processes [10]. Configuration Management is one of the most important processes in ITIL. Through its interconnections with other processes e.g. Change Management, Incident Management, Problem Management etc., it plays considerable role in providing a sound and stable IT service. The Configuration Management Database (CMDB) is in the core of this process and the acronym CMDB is even better known than ITIL [12]. The broad scope of the definitions of the concepts connected with this database, coupled with the fact that there already is an abundance of commercial-grade solutions

which are making an effort to implement it in different ways, only introduce uncertainty and confusion for the people who are trying to introduce a CMDB into their organizations.

ITIL's Versions and the CMDB

ITIL v1 and v2 are focused on the processes that facilitate IT Service Management. ITIL v1 contains thirty books, whereas ITIL v2 only has eight (Service Support, Service Delivery, ICT Infrastructure Management, Security Management, The Business Perspective, Application Management, Software Asset Management, Planning to Implement Service Management) [1], [4]. However, since ITIL v2 is a consolidation of ITIL v1 both versions are largely the same [4]. The Configuration Management process is located in the Service Support section [1]. Version 3 on the other hand focuses on the service lifecycle. It identifies five lifecycle phases – Service Strategy, Service Design, Service Transition, Service Operation and Continual Service Improvement [10]. The Service Asset and Configuration Management process is in the Service Transition Phase [10]. However, no matter which version of ITIL is implemented, the Configuration Management (CM) process and more precisely the CMDB is in the core of the implementation and a deep understanding of all concepts that concern this database is necessary [4].

How has CMDB evolved in order to handle the dynamic IT environment? A comparison between the different versions of the configuration management process can be seen on Table 1.

Table 1. Configuration Management concepts in the different versions of ITIL

	ITIL v1 and v2 [1]	**ITIL v3 [10]**
Focus	Processes	IT Service Lifecycle
CM Process	Configuration Management	Service Asset and Configuration Management
Responsibilities	CM provides a logical model of the infrastructure or a service by identifying, controlling, maintaining and verifying the versions of Configuration items (CIs) in existence.	CM ensures that selected components of a complete service, system or product (the configuration) are identified, baselined and maintained and that changes to them are controlled.

Configuration Item (CI)	Component of an infrastructure – or an item, such as a Request for Change, associated with an infrastructure – that is (or is to be) under the control of CM. Configuration Items (CIs) may vary widely in complexity, size and type, from an entire system (including all hardware, software and documentation) to a single module or a minor hardware component.	Any Component that needs to be managed in order to deliver an IT Service. Information about each CI is recorded in a Configuration Record within the Configuration Management System and is maintained throughout its Lifecycle by Configuration Management. CIs are under the control of Change Management. CIs typically include IT Services, hardware, software, buildings, people, and formal documentation such as Process documentation and SLAs.
Configuration Management Database (CMDB)	A database that contains all relevant details of each CI and details of the important relationships between CIs.	A database used to store Configuration Records throughout their Lifecycle. The CM System maintains one or more CMDBs, and each CMDB stores Attributes of CIs, and Relationships with other CIs.
Configuration Management System (CMS)	A software product providing automatic support for Change, Configuration or version control (Configuration Management Tool is used instead of Configuration Management System).	A set of tools and databases that are used to manage an IT Service Provider's Configuration data. The CMS also includes information about Incidents, Problems, Known Errors, Changes and Releases; and may contain data about employees, Suppliers, locations, Business Units, Customers and Users. The CMS includes tools for managing data about all CIs and their Relationships. The CMS is maintained by CM and is used by all IT Service Management Processes.

Despite the fact that in the third iteration the terms get more detailed there are no considerable differences between the three versions of the IT infrastructure library. Several main conclusions can be drawn from this theoretical survey:

- Configuration Management is not simply a process for recording assets' locations and their accountability – the configuration infrastructure includes items that are outside of the scope of the purely technical aspect of the service and by doing so it poses a considerable obstacle to the person willing to implement a real life software solution. The need for this software system to be able not only to provide accounting information about the assets, but also to acquire new knowledge about the managed service, its configuration,

sustainability, and abilities for future development make it an even harder task. The vital importance of this knowledge extraction through the CMDB is stressed by the introduction of the new Knowledge Management process, introduced in ITIL v3. The low maturity of this process and its vague definition, which sometimes copies the one for Configuration Management, suggests that a sound CM process and CMDB are the keys for the future knowledge discovery and extraction in the ITIL framework.

- The implementation of a successful Configuration Management process starts with the development of a sound CMDB solution – since the database is the core of the CM process its stability is correlated to the stability and robustness of the whole process. The standard process for database development is used to create a sound CMDB solution. It starts with the building of a conceptual and logical model. To enable this the CIs, their parameters and types should be carefully researched, as well as the ways in which they can be interconnected.
- The maturity of the Configuration Management process is dependent on the maturity of the other processes – it is not possible for the CM process to exist in a vacuum and without support from the other processes it cannot reach its full potential. Its closest connections are with the Incident Management, Problem Management, Change Management, Release and Deployment Management, Availability Management, and Finance Management processes. For example, a sound Configuration Management solution would provide Change Management with enough information about what an impact a change would have on the infrastructure. Also, in order for the CI records in the CMDB to be kept up to date a sound Change Management process is required. This holds true for all of the aforementioned symbiotic relationships.
- Although of significant importance Configuration Management is not the universal remedy – judging by the business goals that ITIL v3 outlines for this process [10] one can reach a conclusion that CM is all he or she needs. This, however, is not the case, as can be easily seen by its interconnectedness.

CMDB in Practice

We have already established that the CMDB is not just a database. However, prior to the introduction of ITIL v3 the idea that the practitioners had, which probably stemmed from ITIL v2 [1] was that it was a single database, which in the eyes of the specialists in the field means a single information source, usually of relational origin. An example of such a vision is shown by Spafford [16] who states that the CMDB is a relational database, which serves as a "nerve" center for the management of IT services.

Several authors also agree with the aforementioned statement that the CMDB

is not just a database [9], [12], [13]. This is in line with the vision of ITIL v3, according to which the CMDB is an integrated source of data, which acquires information from many other databases. In order to avoid confusion O'Donnell [13] and also Marquis [9] prefer to use a different term for this vision of the CMDB - Configuration Management System in one case and Configuration Management Databank in the other. In this manner, if the CMDB is considered as one of the numerous databases from which the integrated CMDB acquires its information, then the view for a single relational database, as was the previous idea for the CMDB is completely valid. The same holds true if we think for the integrated CMDB in the terms of Configuration Management System. Marquis [10] even goes as far as to suggest that the CMDB is not just a databank, but also a metabase (metadatabase, metadata repository). He insists that the key to the integrated CMDB is the joining of the metadata for and from all data sources into one database, which in his eyes is the CMDB. If we imagine that there are numerous available data stores, which define the configuration of the IT service, and which can be of various types and granularity, then the CMDB would contain information about all these data stores and would provide the user with an integrated, run-time generated information from them.

These ideas about the CMDB lead to different visions about its implementation too. On one hand, there are those, who like Spafford [16] suggest using the relational model for the CMDB, as it has become, in recent times, a synonym of databases. He supports this suggestion by offering some classification of the most important attributes that a CI has. On the other hand, if we consider the CMDB as an integrated source from disparate sources, it becomes clear that the relational model is not a suitable option for implementation. This is further stressed by the data that is to be saved and extracted – usually connected with different business metrics, such as load, availability, efficiency, effectiveness, etc. All these suggest the usage of the dimensional model, which as Marquis [10] points out is suitable for on line analytical data processing as opposed to the on line transactional data processing typical of the relational model. Both views have been used for real life CMDBs: OpenCMDB is a relational database, and RapidCMDB is an example of a CMDB using the dimensional data model. Which approach to choose depends largely on the organization size and needs, as well as on its capacity and financial conditions.

Conceptual and Logical Model of a CMDB

So far, we have established that the CMDB should save information about the CIs and their interconnections. As simple as it may sound this poses two problems for the implementation:

1. The CIs are of great variety of types and unite different kinds of entities – hardware and software components, people, buildings, documents, etc.

2. The relationships between the CIs are not clearly defined in ITIL and in practice between two arbitrary CIs an arbitrary type of relationship can exist. This poses two additional obstacles:
   a. What kind of relationships should the system support?
   b. How to control the relationships so that only the possible and logical relationships can exist between two specific configuration items?

We tried to find the solutions of these problems in the process of building a conceptual and logical model for the CMDB.

The simplest conceptual model of a CMDB follows directly from the definition of this database [10] and can be seen on Figure 1. It was suggested by Schaaf and Gögetap [15] and although it is of little practical value due to its high conceptual standing it is useful in that it shows visually the definition of the CMDB. Every CI and relationship between CIs has a type associated with it, and every instance of a relationship binds two CIs in a relationship. The beauty of the model is in its simplicity. It is perhaps the only true conceptual model, as every other model adds to it. Lead by this we would look at every other model as a logical one, as it would further define the concepts and by doing so would be of a more concrete and less abstract conceptual level. In order to reach such a model we would address the two issues of the types of CIs and the relationships between them.

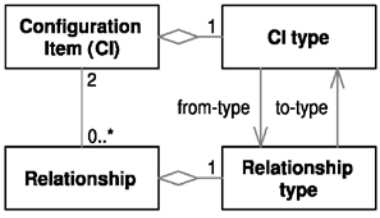

Fig. 1. Conceptual CMDB Model

Every basic classification of CIs identifies physical and logical ones [2], [3]. However, this is too simple and of little practical value. Betz [2] gives another classification, which separates the CI into three classes – CIs, Operational CIs, and Production CIs. These are abstract classes and no object from them can exist. The CI class is the broadest one and every other is inherent from it. The Operational CIs are the CIs involved in day-to-day business processes; they can be measured and are a primary entity in the Service Management workflow [2]. The Production CIs are the ones that are responsible for the actual delivery of the service. As a general rule, in order for a CI record to be in the CMDB it should be important enough to fall under formal Change Management. This classification is more defined but lacks some details that are essential to the modern view of the CMDB. However, it provides for a good basis and we used it for the development of our own classification, which is given on Figure 2.

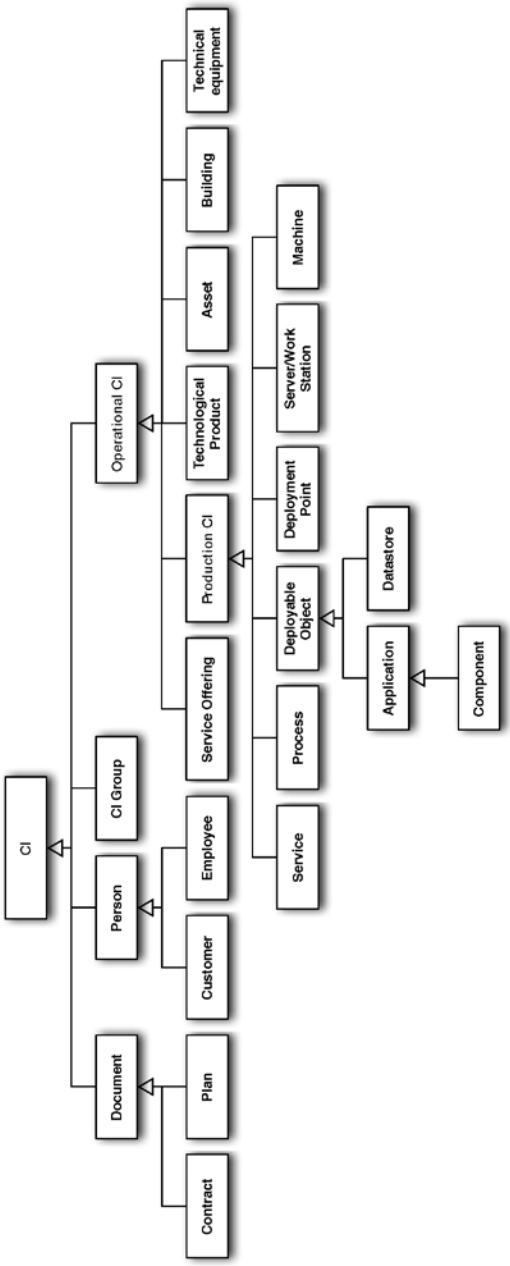

Fig. 2. A Hierarchy of CIs

Several important additions to the model that Betz [2] suggested are made:

- The plan type of document, which is important as it can often include

measurements that can be used as comparison points for different aspects of the service;

- The person type of CI, since according to OGC [14], and McAlpin [11] people involved in the service operation should be treated as a CI. This stems from the fact that personnel is one of the main expenses that an IT organization has and that many problems and incidents may be the result of human error or incompetence. We included two different roles, which a person can have – the customer role and the employee role. These are not necessarily disparate sets, as an employee, regardless of his position in the organization may act as a customer for a system that is hosted by a department different from the one he or she is appointed in.
- The building type of operational CI, which is important as with the joining of the Asset Management process and the Configuration Management process the buildings should be accounted for. The necessity of a separate CI type from the one for asset is suggested by the fact that buildings can be leased and leased objects are not considered assets. Moreover, buildings and any type of space can encompass different physical CIs and as a result need to be managed and controlled, for example, for recovery and contingency measures.
- The technical equipment type of operational CI, which include server racks, disk warehouses, etc. Again, these could be leased and do not count toward traditional assets.

Having reached a hierarchy of the CIs the next remaining question is the possible relationships between the CIs. The main relationships archetypes are dependency, ownership, and enabling. These archetypes can be further decomposed into concrete relationship types as has been done by BMC [3] and Betz [2].

In his model Betz [2] included only two relationship types – ownership and dependence. He also limited the possible relationships between the CIs by allowing only CIs of the same type to exist. This greatly simplifies the model and the implementation. However, we consider that a logical model should be as close to reality as possible so in our model we allow cross-type CI relationships. As a result, the decision about which relationships to be used in a certain system remains a part of its implementation.

Having answered the two main questions we built a logical model for a CMDB with the use of E/R notation, where the entities are represented by rectangles and the relationships' multiplicity is represented by arrows with a straight arrow end meaning one and a trident arrow end meaning many. In order not to crowd the model we have included only the three main archetypes of relationships – dependence, enabling, and ownership. Also, besides the items from the CI hierarchy included are some entities that have not been discussed so far, the examples of which include Program, Project, Release, Problem, (Request for) Change, Incident, Event, Service Request, Risk, and Known Error. Although these entities are primarily the concern of other ITIL processes, without data

and information about them the CMDB would not be complete. Their addition strengthens the integrated nature of the database and makes it more capable for knowledge discovery and mining. The result may be seen on Figure 3.

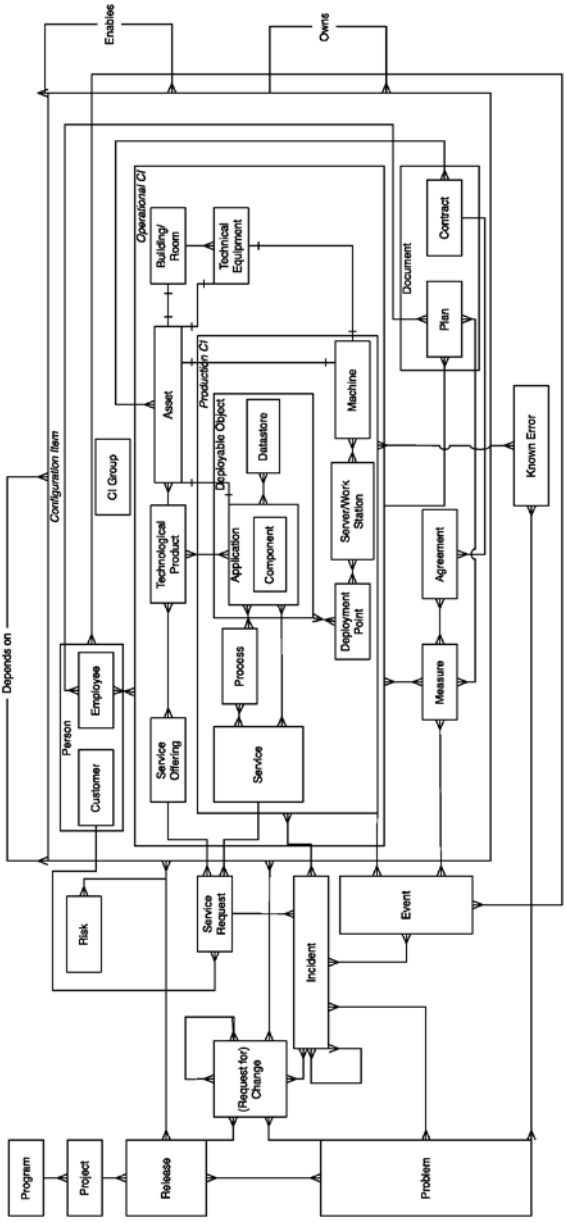

Fig. 3. Logical Model of a CMDB

In order to validate the model, we have developed a test physical model of a CMDB. In it we concentrated on the needs of the small to medium size companies, since there are plenty of commercial grade solutions for the big enterprises (such as BMC's Remedy, IBM's Tivoli Change and Configuration Management Database, Microsoft's System Center and so on). Moreover, the purpose of the creation of the model was to ensure that the concepts we have put into our logical model provide for a sound CMDB solution. The physical model itself implemented only a subset of the logical one, since the needs of the small to medium size businesses are met even without the full scope of the model. We wrote the SQL scripts for the actual creation of the physical model and filled it up with example data, based on real life IT infrastructure details. Then we ran some tests on the database based on real life scenarios. The database proved itself able to handle the necessities of an IT service provider's day-to-day needs, as far as Configuration Management concerns go. It provided some simpler reporting compared to the mentioned commercial solutions, but it was also able to do it at lower production and performance costs and in a more user-friendly way. This also held true for the infrastructure related queries the likes of available storage, infrastructure load, occurring of incidents, etc. These results lead us to believe that the model can be the basis for a full-fledged solution.

Conclusions and Future Work

The building of a logical model is the first step towards the implementation of a sound database solution. The real implementation of ITIL principles and particularly CMDB depends on many details [7] but constructing a model, which is further followed, we confirm once more the significance of the theoretical foundation for improving the modeling process [8], [6]. The model we are suggesting is based on the definitions of ITIL and reflects ideas of various ITIL practitioners. Thus we consider it to be a solid ground on which to elaborate at a later time. However, improvements are always possible.

Additions to the model are possible especially with next revisions of the ITIL. Other sources of such additions are the demands of the real business processes [5]. The new Knowledge Management process is also an interesting source of new requirements for the CMDB since it so heavily relies on it. And knowledge extraction poses quite a few challenges to modern databases. Future work also includes refining the physical model and building working CMDB management software around it.

Acknowledgment. This paper is supported by Sofia University "St. Kliment Ohridski" SRF under Contract 134/2012.

References

[1] Berkhout, M. et al., ITIL: Service Support. [CD edition] London, UK: Crown, 2003.

[2] Betz, Ch. (2005, August 13) A Data Architecture for IT Service Management [On-line]. Available: http://erp4it.typepad.com/erp4it/files/a_data_architecture_for_it_service_management.pdf [May 22, 2011].

[3] BMC Software., Step-by-Step Guide to Building a CMDB, BMC Software, 2008.

[4] Dubey, S. (2008, December 3) ITIL v2 vs. ITIL v3: A Newbie's Guide [On-line]. Available: http://www.saurabhdubey.org/itilv2-v3.pdf [April 1, 2011].

[5] Grigorova K., Process manipulation within DBMS environment, CompSysTech'2002, Sofia, pp.III.28-1-6

[6] Grigorova K. Visualization of database process models, Известия на Съюза на учените - Русе, Серия 5 "Математика, информатика и физика", т. 5, 2005, 67-72

[7] Kaloyanova K., Some aspects of implementing ITIL, Proceedings of the 6-th Annual International Conference on Computer Science and Educaton in Computer Science, 26-29 June 2010, Fulda-Munich, Germany, pp 48-53

[8] Maneva N., Kr. Kraychev, Kr. Manev. Mathematical Models-Based Software Modernization. Mathematika Balkanika J., New Series, vol. 25, No 1-2(2011), pp.131-144

[9] Marquis, H. (2010, April 16) Configuration Management for the Rest of Us [On-line]. Available: http://www.itsmsolutions.com/newsletters/DITYvol6iss15.htm [May 9, 2011].

[10] Marquis, H. (2010, April 30) Enterprise CMDB [On-line]. Available: http://www.itsmsolutions.com/newsletters/DITYvol6iss17.htm> [May 10, 2011].

[11] McAlpin, T. (2007). Can People Be Configuration Items? [On-line]. Available: http://rjrinnovations.com/rjr_docs/Viewpoint Article -Can People Be Configuration Items.pdf [May 27, 2011].

[12] Messineo, D., White Paper: Myths of the CMDB. CA Services. 2009.

[13] O'Donnell, G. (2010, May 10) CMS Pockets of the Truth or CMDB Unified Ambiguity? [On-line]. Available: http://blogs.computerworlduk.com/infrastructure-and-operations/2010/05/cms-pockets-of-the-truth-or-cmdb-unified-ambiguity/index.htm [May 9, 2011].

[14] Office of Government Commerce, ITIL v3: Service Transition, Norwich, UK: The Stationary Office, 2007.

[15] Schaaf, T., Gögetap, B. Requirements and Recommendations for the Realization of a Configuration Management Database [On-line]. Available: http://wwwmnmteam.informatik.uni-muenchen.de/pub/Publikationen/gosc07/PDF-Version/gosc07.pdf [May 20, 2011].

[16] Spafford, G. (2006, April 1) Demystifying the CMDB [On-line]. Available: http://www.spaffordconsulting.com/Demystifying the CMDB.html [May 9, 2011].