CMDB - Yet Another MIB?
On Reusing Management Model Concepts in ITIL Configuration Management

Michael Brenner, Markus Garschhammer, Martin Sailer, Thomas Schaaf

Munich Network Management Team
University of Munich
Oettingenstr. 67
D-80538 Munich, Germany
{brennera|garschha|sailer|schaaf}@mnm-team.org

Abstract. According to ITIL, a CMDB (Configuration Management Database), containing a logical model of the IT infrastructure, forms the basis for effective and efficient IT Service Management. However, a common understanding of what constitutes a CMDB has not yet been established. By contrast, concepts for building and using MIBs (Management Information Base) – also aimed at providing logical models of the IT infrastructure – have long since been established in the area of systems management.
This paper presents an overview of the CMDB and MIB concepts, discusses how they relate to each other and compares them based on the main purposes of a CMDB. It discusses whether modeling approaches used for MIBs can be reused for CMDBs. To this end, a criteria catalog based on core CMDB concepts and basic information requirements of ITIL's Service Management processes are derived, and the challenges of implementing a CMDB reusing concepts of common management models are discussed. Concluding, basic approaches towards integrating CMDBs and MIBs are presented.

1 Introduction

In approaching ITSM *(IT Service Management)* issues, there is a current trend towards greater consideration of organizational (rather than purely technological) aspects. In this context, the *IT Infrastructure Library* (ITIL) has, of all standardization efforts, gained the biggest popularity and can – at least in Europe – now indeed be called a de-facto standard. In its core titles *Service Support* and *Service Delivery*, ITIL provides "best practice" guidelines for IT Service Management.

Implementing *Configuration Management*, a central process of Service Support [1], is often considered the biggest stumbling block in ITIL realization. This is not just because the Configuration Management process itself is not as structured as the other Service Support processes [2]. It is defining the scope and structure of the CMDB, as well as filling and maintaining it, which often proves to be exceedingly difficult and time-consuming. Thus, one of the main challenges in developing solutions for supporting the application of ITIL is to detail the CMDB concept in order to facilitate its implementation.

In the area of systems management, however, integration efforts have given rise to a number of standardized information and data models. Essentially, these models provide means to build a *Management Information Base* (MIB) to be used by IT infrastructure management systems – i.e. a MIB is a logical view on the management-relevant aspects of a part of an IT infrastructure. Thus, at first glance, a CMDB could be seen as not much else than yet another MIB (or a collection of MIBs).

The remainder of this paper is structured as follows: Sec. 2 defines the usage of important terms in the context of this paper and illustrates correlations between the terms used in a CMDB context and those used in a MIB context. This leads to the interesting question: Are existing techniques (management models) capable of solving some of the problems occurring when setting up a CMDB? The answer is based on examining core CMDB concepts (Sec. 3), defining basic criteria for a potential CMDB model, and applying these criteria to some of the most common management models (Sec. 4). Sec. 5 concludes by further investigating the commonalities and differences of the concepts underlying CMDBs and MIBs and discussing possible integration approaches.

2 From Managed Objects to Configuration Items

When discussing management models or ITIL Configuration Management, many terms mean different things to different people. The following will define the usage of some essential terms for the context of this paper.

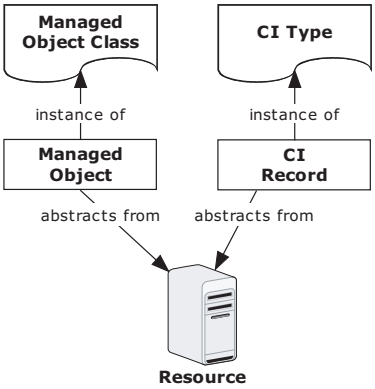

Fig. 1. CIs vs. MOs

The *Managed Object* (MO) concept has been defined in the OSI Management Framework [3] and been referred to in various architectures for network and systems management. MOs represent the management view of resources, i.e. they abstract from resources (components) in a managed IT infrastructure. They can be seen as an abstract model of a resource or as the data record used to express this model[1]. The set of MOs associated with a system constitutes that system's *Management Information Base* (MIB). MOs that share the same definition are instances of the same *Managed Object Class* (MOC) [4].

Basically, a management *Data Model* (DM) contains MO definitions (or MOCs) in a formalized and detailed enough way to enable a straightforward MIB implementation. Consequently, a DM is bound to a Data Model Language (DML) which defines the syntax in which MOCs are described. If the DML prescribes *how* management information is represented in the DM, the *Information Model*[2] (IM) defines *what* this management information should comprise. In other

[1] In common usage, the term "MO" can also denote the managed component itself.

[2] In the context of this paper the term "Management Model" will be used to refer to the composite model that the union of IM, DML and DM builds (cp. Sec. [5]), but note that in a broader context this is often referred to as the "Information Model" of a management architecture [6].

words, a DM is formalizing and detailing the concepts contained in an IM – consequently various DMs can be derived from a single IM. In an IM context an MO would be an abstract model – in a DM context, an MO would be a data record representing that model. Analogous, in an IM context, MOC is the abstract model of a class of MOs (or a type of MO) – in a DM context, an MOC is a formal definition that is used to instantiate MOs of a specific type (or used to be refined into other MOCs). Most of the standardized management models focus on defining a DM, and document the underlying IM only informally or incompletely [5].

The concept of a *Configuration Item* (CI) in the context of ITIL's guidance on *Configuration Management* (CM) seems similar to that of an MO. A CI is a component of an IT infrastructure (or other items associated with that infrastructure) which is put under the control of the Configuration Management process [1]. Relevant information on CIs (CI attributes) and the relationships between CIs are to be recorded in the *Configuration Management Database* (CMDB). ITIL does not make a clear distinction between the CM view (model) of a component or its expression in the documented CI attributes and the component itself. In this paper, the term *CI Record* (CIR) will be used to refer to the data record contained in the CMDB, while CI will denote the abstract model (CM view) of the component. As depicted in Fig. 1, a CI is abstracting from a resource much in the same manner as an MO. According to ITIL, CIs should be classified into *CI types*. ITIL takes a broader view on what should be regarded as a CI (including e.g. documentation and services defined in SLAs) than what is usually referred to as an MO. Also a CMDB seems to cover a larger part of an IT infrastructure than what one usually thinks of as a "system" (the scope of the management information stored in a MIB). However, since CI records often describe the same kind of IT resources (like software and hardware items) as MOs, there is an obvious analogy between the concepts of MO and CI/CIR, MOC and CI type as well as MIB and CMDB.

Given the existence of standardized management models for network and systems management that define a DML, as well as DMs with large numbers of pre-defined MOCs, it surprises that equivalent standards for building CMDBs have not been proposed so far. This begs the question whether some of the existing management models might be useful for filling this gap. Before this, the concept and purpose of a CMDB needs to be analyzed in more detail.

3 The CMDB Idea

Unfortunately, the guidance ITIL itself gives on the CMDB is neither comprehensive nor consistent in all details. Consequently, it is difficult to give a compendious definition of it. Basically there are two views on the CMDB within ITIL which are not necessarily conflicting, but still address distinct aspects. On the one hand, the CMDB is a logical model of the IT infrastructure and IT services whose creation and maintenance is the main deliverable of the *Configuration Management* process, as discussed in the according chapter in Service Support [1]. On the other hand, the CMDB is seen as a sort of "information hub": In most cases, whenever an ITIL service management process needs to access information outside its immediate scope of responsibility, this is supposed to happen through querying the CMDB. This information can refer to things

quite different from IT infrastructure elements or services, e.g. artifacts of other ITIL processes like incident records, but also records on information like customer and user data, whose control is usually not within the scope of IT management. ITIL itself makes no definitive statements on this duality of the CMDB (or even acknowledges it expressively), but it is up to discussion in any real-world implementation whether Configuration Management needs to maintain the latter kind of information – or just ensure access to it. Besides demanding the documentation of relations and dependencies between CIs (including containment relations), ITIL gives almost no guidance on CMDB implementation specifics.

For Configuration Management ITIL defines five[3] basic activities [1,7]: *Configuration Management Planning* (defining scope, purpose, responsibilities etc. of Configuration Management), *Configuration Identification* (defining what CIs are to be included into the CMDB and in what form), *Control of CIs* (assuring up-to-date recording of the characteristics of CIs – in particular in the event of changes), *Configuration Status Accounting* (reporting and control of the current version and change history of all CIs) and*Configuration Verification and Audit* (Planning and carrying out configuration audits to verify accuracy and completeness of the CMDB).

The first two activities listed could be seen as parts of a CMDB setup project. In the terms of Sec. 2 main tasks of such a project would include defining an IM for the CMDB, choosing a suitable DML and a DM. Note however that *Continuous Improvement* is a central concept underlying all ITIL guidance, and consequently, these two activities might be repeated in its context. An important conclusion can be drawn from this: The requirements for the information contained in the CMDB are scenario specific (IT organization specific), can change, and consequently the IM underlying a CMDB might need to be adapted (see also Sec. 4.1). This does not mean however that the requirements on the IM will differ vastly between two providers of similar IT services – there should be large intersections that could be used for building "Base Information Models" for CMDBs.

"Configuration Status Accounting" is a concept adopted from Software Configuration Management that refers to keeping track of the life cycle status of a CI (what version of a specific CI is "in testing", "in operation" etc.). This implies that infrastructure elements should be tracked in a CMDB even before they come into operation. Also, as can be seen from the activities "Control of CIs" and "Configuration Verification and Audit", the concept of ITIL Configuration Management per se assumes more or less manual maintenance of the CMDB and does not rely on any technology to feed data into it. In a ITIL-aligned organization this is not necessarily as complex as one might assume. In day-to-day operations, the only modifications to the content of a CMDB should be triggered by activities of the Change Management process. Results from automatic infrastructure scans can be extremely useful for "Configuration Verification and Audit", but should not be fed unreviewed into the CMDB. Still, this implies that the practice of CMDB maintenance stays comparatively labor-intensive and rises with the amount of information stored in the CMDB. It is therefore important to find the appropriate level of detail in which CIs are to be recorded in the CMDB, achieving a balance

[3] seven on some counts, but we disregard like most other literature the very generic "CMDB back-ups, archives and housekeeping" and "Providing a Configuration Management service"

between the benefits of information availability and the resources and effort needed to support it [1]. So in summary, a CMDB

- exits to serve the essential information needs of the ITSM processes defined in ITIL
- should be kept "slim" and closely aligned with these information requirements
- contains a model of the IT infrastructure and services
- documents relations between any CIs.

But even though the concept of a CMDB is different from that of a MIB (see Sec. 5), at their core they still both model (parts of) IT infrastructures. ITIL neither gives concrete instruction towards implementing a CMDB, nor have standardized models been established. This begs the question, whether existing information models can be used or adapted for building CMDBs.

4 On Reusing Management Models

Even in the few instances where criteria for CMDB tools are discussed, these efforts are usually focussed on functional requirements (e.g. visualization) and integration with other databases [8,9]. Limiting assessments only to these requirements however bears the danger of not addressing key standardization issues for CMDBs. An effective and sustainable solution to CMDB integration issues will need to be based on at least some partial standardization on the level of a CMDB IM, DM and DML. Consequently, the criteria outlined below are a first step towards documentation of requirements for the design (or selection) of an IM, DM and DML that a CMDB tool will explicitly or implicitly have to be based upon. In Sec. 4.2 the proposed criteria are applied to three existing management models to evaluate the possibilities of reusing their concepts.

4.1 Requirements on a management model reusable for building a CMDB

With the CMDB concept being rather ambitious, a management model for a CMDB will need to fulfill a number of requirements. Note that many requirements cannot be exactly mapped to either DM, DML, or IM, as there are many interdependencies between these aspects (e.g. a very simple DML might not be able to express complex IM concepts) as well as between them and implementation or architecture specifics (e.g. will the CMDB comprise several physical databases, has Configuration Management a say in how information is stored and accessed in other enterprise databases?).

Adaptability of Model - All ITSM processes are subject to *Service Improvement Programs* (Continuous Improvement, cp. Sec. 3). Consequently, the CMDB, subject to *Continuous Improvement* as well, must be capable of dealing with changing requirements, especially regarding scope, nature and level of detail of the documented information. To keep the costs of adapting the IM low, the DML should allow easy extensibility of the DM.

Alignment to ITSM information needs - Obviously, the IM for CMDB should address all the information requirements of the ITSM processes and consequently include

models of all relevant entities (including e.g. Incident Records etc.). On the other hand, to keep the CMDB in principle "human maintainable", it should not cover too much information or aspects which are not essential in this context.

Comprehensive view on infrastructure and component relations - The documentation of CI relationships (e.g. for service impact analysis) is maybe the single most essential concept in the CMDB context. Consequently, the IM should include basic relations between common CI types and the DML should support modeling multiple, preferably even user-definable, relationships between CIs.

Inclusion of ITSM process artifacts - This criterion refers to the "information hub" nature of the CMDB (cp. Sec. 3). Each ITSM process defined in ITIL not only has specific requirements about what information should be contained in a CMDB, it also creates (ITSM) data itself, i.e. process artifacts such as incident records. Analogous to above criterion, the relationships among these process artifacts (e.g. what incident records are linked to a specific problem record?) as well as between them and infrastructure CIs (e.g. what problem records are associated with a specific infrastructure CI?) need to be included in the CMDB and in consequence should be part of the IM. As there is no convincing argument for putting all process artifacts under Configuration Management control in the same way it is done with infrastructure CIs – and these process records are usually controlled through process-specific tools (e.g. Incident Management System) – it seems likely that in a real-world CMDB implementation these artifacts and infrastructure CIs will be stored in separate physical databases. In that case, a common DML suited for building models of infrastructure components as well as process artifacts could ensure obstacle-free integration.

Integration with external databases - Information of relevance for ITSM might be managed and stored outside the IT organization itself – either in enterprise databases (for example employee data managed by the Human Resources department) or in external CMDBs (e.g. of an external IT sub-service provider). This information will most likely be outside Configuration Management control, and consequently so will be the technical nature of access methods. The DML and DM should therefore lend themselves to integration with a variety of data sources (e.g. by providing easy XML/Web Services mappings).

Integration with (other) network and systems management data stores - Much of the information that should be contained in a CMDB cannot be gathered solely by using resource management or discovery tools (e.g. systems' locations, compositions of services). However, for information aspects that can be discovered using management tools, verification and audit of CMDB records can benefit greatly from integration with these tools. A DML for Configuration Management should therefore ease reconciliation of data stored in the CMDB with that of other existing management systems (e.g. by providing mappings to other common DMLs).

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 IM. Also information pertaining to all life cycle phases should be accessible through a CMDB – via attributes or relationships to other CIs and data records (e.g. acquisition date, test records, etc.).

Catalog of basic CI types - Provisioning of common CI types (or MOCs) (informally in the IM – though preferably in the form of an extendable but ready-to-use DM) could significantly shorten the time-to-implementation for a CMDB.

4.2 Assessment of current management models

The apparent similarity of the MO and CI concepts prompts for a reexamination of existing management models. By applying the criteria catalogue presented above, we assess state of the art management models regarding their possible reuse building CMDBs.

Fig. 2 gives a comprehensive overview of the criteria catalog and illustrates how different approaches fulfill these criteria. In the following, we discuss the *Internet Management Model* (IMM) [10], the *Common Information Model* (CIM) [11] and the *Shared Information/Data Model* (CIM) [12].

Internet Management Model (IETF) The Internet Management architecture has two main pillars: The *Simple Network Management Protocol* (SNMP) and a large number of *MIB modules* published in RFCs [5]. The latter build what, though there is no official name for it, could be called the *Internet Management Model* (IMM) that covers a lot of system types in its scope. Following its original design goal of providing a simple way to manage network resources, this model's focus is comparatively narrow. MIB modules contain the MO definitions for a specific type of system – but in IMM an MO often represents a very small aspect of a system that one would generally rather think of as an attribute, e.g. an MO can be a single counter variable. In the Internet Management architecture MOs/MIBs are intended to be stored on the managed system and accessed remotely via an SNMP agent. In the terms defined in Sec. 2, a MIB module (e.g. for a type of switch) could be seen as a data model MOC representing a type of system.

Documenting relationships between MOs is not supported by IMM (except containment within a single system's scope) and as an IMM-MIB is limited to describing a single system, a view on the entire infrastructure and the relationships between its components is not supported by the model. In practice this gap is often filled by functionality provided by SNMP-based network management tools (management platforms) that for example support viewing network topologies. Also, an Internet MIB is concerned only with the operational state of a resource. IMM is not designed to support tracking the lifecycle status of a resource.

With the Internet Management architecture being the first widely adopted and implemented management standard, integration with other standards was at the time of its conception of no concern – though concepts for integrating it with later management architectures have been designed. Also, due to its technical focus, support for integration with enterprise databases or linking to documents (process artifacts) was never intended in IMM. The concept of IMM presumes the requirements of management to be rather static. It is not intended that MIB modules can be customized by the user (i.e. the operator of the infrastructure), e.g. for addressing operator-specific or changing requirements.In practice, it is again management platforms that address this gap by filtering or consolidating information or allowing to retrofit the infrastructure view gathered from the MIBs with manually added information.

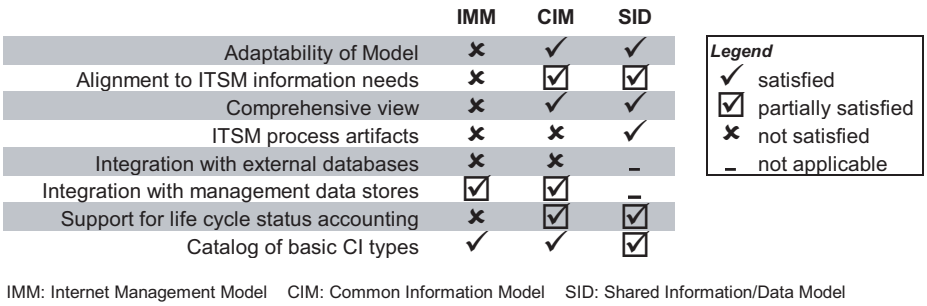

Fig. 2. Assessment of management models

Common Information Model The Common Information Model (CIM) [11] is an object-oriented management model that aims at providing a common way to represent information about networks and systems as well as services. It defines managed resources as object classes that can be further refined by means of strict inheritance. Part of CIM is textual, human-readable language (*Managed Object Format (MOF)*[13]) for describing modeling constructs that can be processed by automated tools. Since all CIM classes derive from the `managed element` class as defined in the Core Model, CIM provides a coherent view on the modeled infrastructure. This view, however, does not include the linkage of ITSM processes to infrastructure elements. In particular, key concepts of ITIL such as Incident records fall out of CIM's scope.

CIM makes extensive use of relationships, namely associations and aggregations; together with its object-oriented approach this yields a sufficient level of expressiveness and extensibility. CIM features a large amount of standardized object definitions. In total, the amount of managed object classes defined in CIM comes close to 900. It is therefore fair to say that CIM represents a solid basis for integrated management, but is a fairly complex model. Understanding the relationships between these classes and adapting CIM to the needs of an organization requires a serious effort [14]. Moreover, much of the information conveyed in these classes deals with low level details of resources and clearly exceeds the ITSM scope.

Despite some exceptions within the CIM application schema, CIM's focus is on the operation phase of an IT infrastructure. Thus, it provides only rudimentary support for life cycle phases such as planning.

While CIM provides a rich data model, it currently shows deficits in expressing business-oriented concepts as required by ITSM. In particular, only few MOCs for expressing service-related management information have been defined so far. This might be due to the fact that it has its roots in the area of desktop systems and has over time developed into a more generic model.

Shared Information/Data Model Compared to CIM and IMM, SID (Shared Information/Data Model) [12,15] is less centered around DM concepts, but provides an information model. SID is an integral part of the NGOSS (New Generation Operations Systems and Software) initiative by TMF (TeleManagement Forum). Including SID

in this assessment thus bears some ambiguities. However, SID is the first information model tightly coupled to management processes. If there was a CMDB for the eTOM process framework, it would be based on SID.

The SID model employs an object-oriented modeling approach and draws a clear distinction between the system and business view on management information. Accordingly, it is organized into System and Business domains, which are in turn partitioned into Aggregate System Entities (ASEs) respectively Aggregate Business Entities (ABEs). ASEs are intended to facilitate linkage between business and system view, since they elaborate on the concepts defined in ABEs. SID is strongly tied to the eTOM Process Framework [16] in that Business Domains accord with eTOM Level 0 concepts. However, since eTOM and ITIL vary considerably in structure, SID's ABEs will not be suitable for unqualified inclusion into a CMDB information model.

The concept of *Continuous Improvement* is not explicitly addressed in eTOM and NGOSS. It is therefore not quite clear yet wether user/operator adaptability will be a central design goal for future implementations of a SID data model. Inherently, the SID model is specified as a rooted class hierarchy. It makes use of object-oriented concepts like generalization, associations, aggregations and compositions to express relationships between entities and is thus able to provide a coherent view on the IT infrastructure.

With entities and attributes being described by a mixture of descriptive text, UML diagrams and tables, SID also provides a reasonable level of expressiveness. This includes the use of finite-state machines to model life-cycle aspects – a concept that has been incorporated from DEN-ng.

SID's strength clearly lies in its modeling of higer-level concepts (e.g. *Service*, *SLA*), where most MOCs have been defined. While in this regard SID offers considerable benefits over IMM and CIM in terms of maturity, it currently defines only few MOCs for expressing low level details of resources. While SID's focus is clearly on eTOM processes, it exhibits a number of sound concepts that a CMDB information model would benefit from. This includes the coupling of model entities and business processes to provide a business viewpoint on the data/information.

The detailed investigation of three different concepts for information modeling showed that none of them is directly applicable to build a CMDB. For instance IMM and CIM do not offer any support to model the life cycle dependence of CIs. All approaches lack a strict focus on ITSM. Besides SID, no information model offers standard, easy to deploy MOCs for expressing higher-level concepts such as services. However, it shows deficits in modeling low-level parameters of system and network components – whereas CIM and IMM feature a large amount of standard MOCs for that purpose. Fig. 2 comprehensively illustrates these gaps towards implementing a CMDB based on well established information models.

5 The CMDB-MIB Gap and First Steps on Bridging It

As seen in Sec. 4.2, the established management models do not lend themselves to immediate application for CMDB design. The reason for this apparently lies in some fundamental differences between the design goals and design philosophies for MIBs

and CMDBs. The purposes a CMDB and a MIB serve are, despite seeming similar on the surface ("providing a model for IT management"), quite distinct. Some distinctions were already touched upon in previous sections. This section outlines some additional aspects before continuing to discuss approaches to integrate MIBs and CMDBs.

5.1 Understanding differences and similarities – the CMDB-MIB gap

A MIB serves to make the tasks of day-to-day operations easier for operators and administrators by addressing the challenges of infrastructure diversity and (physical/geographical) distribution. In contrast, a CMDB serves decision makers in the various ITIL service management processes for which a comprehensive overview (e.g. on the factors influencing service performance) is often more valuable than minute detail.

Also the philosophy of who has the last say about the contents of a CMDB or MIB differs. A MIB usually comes with the system it models, i.e. its design is done by the system vendor. A later customization by (management) users on a model level is not intended. Realizing adapted models is often possible through functionality of management systems – these are then usually stored in a format proprietary to that management system. ITIL's "adopt and adapt" philosophy by contrast suggests that the scope and structure of a CMDB should be adaptable to cope with changing scenario-specific requirements.

Also, while tracking components through all life cycle phases, CMDBs do not provide up-to-the-minute data of the operational status of components – nor is a CMDB intended to be a tool to manipulate that status with effect onto the live environment. Even though editing and auditing the CMDB can be made more efficient by tools, the maintenance of the CMDB as a whole is an organizational practice independent from any technology, and can consequently probably never be 100% automatized.

So, a CMDB is really a different beast than a MIB. Both emphasize in their models distinctly different aspects of the IT infrastructure as they serve to support different tasks by different groups of stakeholders. But then again, they both offer an abstracted management view of the same infrastructure.

There is an analogy to that in IT system modeling. It is commonly accepted that there is a need for different model types in software and systems engineering. As it is good practice in this discipline, there should be mechanisms for ensuring consistency between MIB and CMDB models and facilitate reuse of shared information.

5.2 Further directions – bridging the CMDB-MIB gap

Some ITIL consultants might argue that the requirements for the model underlying a CMDB are dependent on the individual characteristics of every IT organization, and therefore no common model will fit all scenarios. However, the growing adoption of ITIL (and related standards like ISO 20000) furthers a basic organizational standardization – while on the infrastructure side, standardization of hardware and software has also been evolving. It therefore seems very unlikely that the requirements for a CMDB will vary so enormously across IT organizations that are comparable in size and offered services. Given the very high level of abstraction and generality in ITIL's CMDB guidance, there is ample room (and need) for a common CMDB "reference model",

comprising an IM, DML and DM that are applicable to a large number of CMDB scenarios (and adaptable to suit most).

While a sound approach for such a standardized model for ITIL is currently missing, the TMF has pursued a similar goal with SID (cp. Sec. 4.2). Although SID is built around eTOM processes and therefore not directly applicable to ITIL-based ITSM, many of its concepts seem to be general enough in design to be suitable for reuse in CMDB models.

Towards this goal, we are working on the definition of a CMDB reference model. The first step is to define a set of CI types which are indispensable for ITIL-based IT Service Management and the relationships between them. (e.g. *ServiceIncident, ServiceProblem*). When defining CI Types for common kinds of infrastructure components or business entities, we try to reuse existing (abstract) information model concepts from CIM and SID. Information items that have not been covered by these approaches, such as ITIL process artifacts, are conceptualised using reference processes we concurrently develop based on ITIL process descriptions (cp. [2]). In particular, this includes the information flow between processes – an aspect only incompletely addressed by ITIL for some processes – and basic *Key Performance Indicators* (KPIs).

Given the differences between MOs and CIs, it seems likely that MIBs and CMDBs will coexist in most IT organizations in the foreseeable future. It would be advantageous to have means of exchanging information between them, e.g. for facilitating CMDB verification and audit. After all they both contain models of the same infrastructure. For this, the ability to define mappings between the MIB and CMDB data models (necessarily based on IM and DML mappings) would be desirable. But we hope that also information flow in the opposite direction could be an asset. If service dependencies maintained in a CMDB can be integrated with the real-time data out of MIBs, this might be a step forward towards the ambitious goal of tracking the operational status of a service in a "Service MIB" (cp. [17]).

6 Conclusion

This paper presented a comparison of the common notion of a MIB against the yet evolving idea of a CMDB. An assessment of established management models showed that their concepts cannot be reused unadapted for building a CMDB. The cause for this lies in a fundamental difference between the principles underlying both concepts, stemming primarily from the distinct goals pursued in their design.

In the light of ITIL-based ISO 20000 certifications for IT service providers gaining momentum, an approach for building capable and sustainable CDMBs is dearly needed. This is achieved best not by retrofitting existing tools, but by approaching CMDB design "top-down" – based on a sound requirements analysis and the development and standardization of appropriate models. In addition, given the existence of MIB-based management systems in most larger IT organizations, there is a strong point for arguing that a CMDB should be closely integrated with these. Approaching such an integration should start at the model level as well and not be based on "bottom-up" application integration.

Acknowledgment
The authors wish to thank the members of the Munich Network Management (MNM) Team for helpful discussions and valuable comments on previous versions of this paper. The MNM Team directed by Prof. Dr. Heinz-Gerd Hegering is a group of researchers of the University of Munich (LMU), the Technical University of Munich (TUM), the University of the Federal Armed Forces Munich, and the Leibniz Supercomputing Center of the Bavarian Academy of Sciences. Their web-server is located at http://www.mnm-team.org. This paper was supported in part by the EC IST-EMANICS Network of Excellence (#26854).

References

1. Office of Government Commerce (OGC), ed.: Service Support. IT Infrastructure Library (ITIL). The Stationary Office, Norwich, UK (2000)
2. Brenner, M.: Classifying ITIL Processes — A Taxonomy under Tool Support Aspects. In: First IEEE/IFIP Workshop on Business–Driven IT Management (BDIM 06), IEEE (2006)
3. ISO/IEC: Management Framwork. (1989) ISO 7498-4.
4. ISO/IEC: Management Information Model. (1993) ISO 10165-1.
5. Pras, A., Schoenwaelder, J.: On the difference between information models and data models. Technical report, IETF (2003) RFC 3444.
6. Hegering, H.G., Abeck, S., Neumair, B.: Integrated Management of Networked Systems. Morgan Kaufmann Publishers (1999)
7. OGC, ed.: Introduction to ITIL. IT Infrastructure Library. The Stationary Office (2005)
8. Colville, R.J.: CMDB or Configuration Database – Know the Difference. Gartner. (2006) Research Note G00137125.
9. Pink Elephant: PinkVerify – Configuration Management Certification Criteria. (2006) http://www.pinkelephant.com/en-GB/PinkVerify/.
10. Case, J., McCloghrie, K., Rose, M., Waldbusser, S.: RFC 1902: Structure of Management Information for SNMP V2. (1996)
11. DMTF: Common Information Model (CIM) Version 2.9. (2005)
12. TMF: Shared Information/Data (SID) Model Concepts, Principles, and Domains. TMF. (2006) GB922.
13. DMTF: Core MOF Specification 2.1. (1998)
14. Alexander Keller, Heather Kreger, K.S.: Towards a CIM schema for runtime application management. In: International Workshop on Distributed Systems: Operations & Management (DSOM 2001), IFIP/IEEE (2001)
15. Strassner, J.: Policy based network management. Morgan Kaufmann Publishers (2004)
16. TMF: enhanced Telecom Operations Map (eTOM), The Business Process Framework For The Information and Communications Services Industry. (2002) GB921.
17. Sailer, M.: Towards a Service Management Information Base. In: IBM PhD Student Symposium at ICSOC05, Amsterdam, Netherlands (2005)