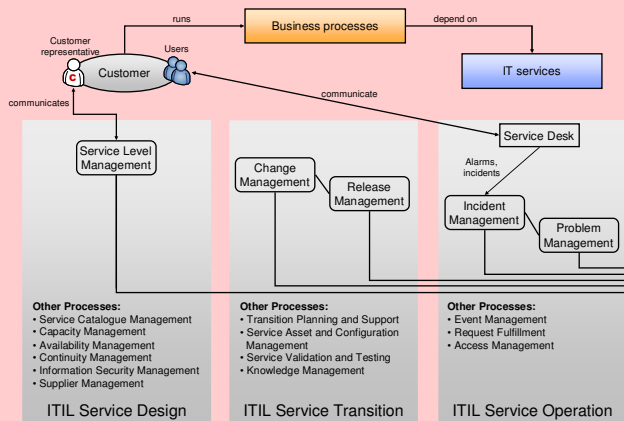


Requirements and Recommendations for the Realization of a Configuration Management Database (CMDB)

Challenges in CMDB Realization

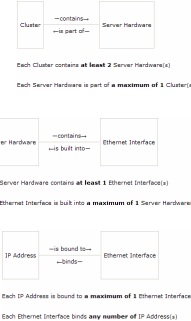


Requirements on a CMDB

Information Model Requirements

- 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 (CRs), etc.).
- IM3 Comprehensive view on 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 CI types or patterns**
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.

Examples of Configuration Management Patterns



Recommendations for CMDB Realization

- 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 multiple data confirmation concepts**
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 cardinalities and 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 an 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 layered visualization**
The visualization (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.

Prototypical Implementation of a Customizable CMDB Tool

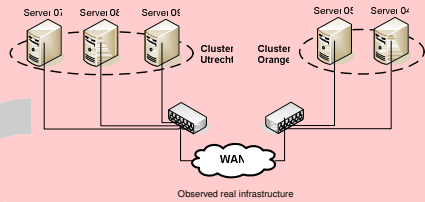
The screenshots show a prototypical implementation of a customizable CMDB tool. It includes a table for configuration management, a flowchart for relationship management, a task list for cluster change, and a table for cluster change. The tool is branded with BPMSpace.

I The CMDB as a Service Knowledge Management System

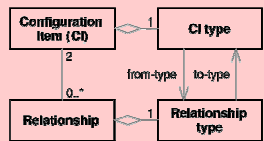
The CMDB serves as an "Information Hub" between IT Service Management processes.

II The CMDB as an Infrastructure Modeling System

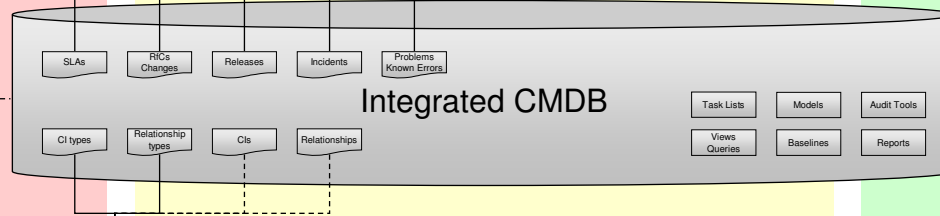
- The CMDB contains a logical model of the IT infrastructure and services.
- Configuration Items (CIs) represent models of resources and services.
- The CMDB documents relationships between CIs.



- CI types:**
 - Cluster
 - Server
 - Ethernet Interface
 - IP Address
 - LAN
 - WAN
 - ...
- Relationship types:**
 - Cluster contains Server
 - Server is part of Cluster
 - Server contains Ethernet Interface
 - Ethernet Interface is installed in Server
 - IP address is bound to Ethernet Interface
 - Ethernet Interface has IP address
 - ...

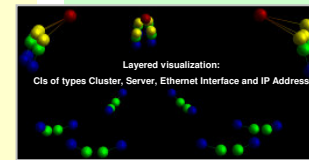


Integrated CMDB



Functional Requirements

- F1 CI Discovery support**
The discovery of CIs requires adequate support by the CMDB tool, in particular with respect to naming conventions, data consistency 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 as 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 data information sources and tools**
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.



3D Graph Visualization

