

# An Overview of the Common Model Library for IT Management

A Joint White Paper from BEA Systems, BMC Software, CA, Cisco, Dell, EMC, HP, IBM, Intel, Microsoft, and Sun Microsystems

Version 1.0, Date 12/13/2007

## Abstract

The Common Model Library (CML) defines common expressions of, and the semantics for, concepts that enable information exchange between management tools. The specification for CML will be expressed using the Service Modeling Language (SML).

Some of the benefits of the use of CML are to facilitate automation, simplify the management and deployment of services, and reduce the cost of integrating new management tools and managed resources into an IT system.

## Authors

Wayne Adams, EMC  
John Arwe, IBM  
J. Keith Bankston, Microsoft  
Jordan Boucher, Sun Microsystems  
Don Deel, EMC  
Zulah Eckert, BEA Systems  
Patricia Florissi, EMC  
Paul Gleichauf, Cisco  
Steve Jerman, Cisco  
Vincent Kowalski, BMC Software  
Heather Kreger, IBM  
Phillip Liu, HP  
Milan Milenkovic, Intel Corporation  
Bryan Murray, HP  
Vijay Tewari, Intel Corporation  
John Tollefsrud, Sun Microsystems  
Marv Waschke, CA  
Andrea Westerinen, Microsoft  
John Wilson, Dell

## **Copyright Notice**

Copyright 2007 by BEA Systems, Inc., BMC Software, Inc., CA Inc., Cisco Systems, Inc., Dell Inc., EMC Corporation, Hewlett-Packard Development Company, IBM Corporation, Intel Corporation, Microsoft Corporation, Sun Microsystems, Inc. All Rights Reserved.

THIS WHITEPAPER IS PROVIDED "AS IS," AND THE AUTHORS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THIS WHITEPAPER ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

THE AUTHORS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THIS WHITEPAPER.

The name and trademarks of the Authors may NOT be used in any manner, including advertising or publicity pertaining to this Whitepaper or its contents without specific, written prior permission. Title to copyright in this Whitepaper will at all times remain with the Authors.

No other rights are granted by implication, estoppel or otherwise.

## **Status**

This whitepaper is provided for review and evaluation only.

At some future date, the contents may be published under another name or under several new specifications, as shall be agreed by the authors and their respective corporations at that time.

# Table of Contents

<b>An Overview of the Common Model Library for IT Management .....</b>	<b>1</b>
Abstract .....	1
Authors .....	1
Copyright Notice.....	2
Status.....	2
Table of Contents .....	3
<b>1. Introduction .....</b>	<b>4</b>
<b>2. Scenarios .....</b>	<b>5</b>
2.1 Impact Analysis .....	5
2.2 What-If Analysis .....	5
2.3 Service Deployment .....	6
2.4 Configuration Management Database.....	6
<b>3. Business Goals .....</b>	<b>7</b>
<b>4. Model Library Overview.....</b>	<b>7</b>
4.1 Model Library .....	8
4.1.1 Common Language.....	8
4.1.2 Consistent Information Encoding .....	8
4.1.3 Common Elements .....	8
4.1.4 Shared Elements .....	9
4.1.5 Semantic Definitions .....	9
4.2 Additional Design Principles and Goals .....	9
4.3 Non-Goals .....	9
<b>5. CML and SML .....</b>	<b>10</b>
<b>6. Deliverables .....</b>	<b>10</b>
<b>7. Future and Next Steps.....</b>	<b>11</b>
<b>8. References .....</b>	<b>11</b>

# 1. Introduction

Today's CIO is looking for ways to drive down IT Total Cost of Ownership (TCO) while realizing more predictable and business-aligned outcomes from IT. Developing and deploying new business services is costly and also time consuming. Lower IT TCO can be enabled through greater integration of management tools and managed resources, and through increased automation of IT processes.

End-to-end management of a complex enterprise infrastructure of people, processes, information, and technology must span a broad range of management disciplines and lifecycle phases. Management disciplines typically include performance, service desk, configuration management, systems management, network management, change management, service level agreement (SLA) management, asset management, and more. There are many lifecycles in IT management, for example IT service and operations lifecycles. Lifecycle phases typically include the planning, development, deployment, and maintenance of an IT service. End-to-end management requires communication across a variety of management tools and managed resources. Since these tools and resources often were developed by different vendors using different technologies and data models, the exchange of data often requires the pair-wise implementation of adaptors, translators, and transformation code to bridge the gaps in syntax and semantics. Human intervention is often required in the process of translating between management tools and managed resources which drives up both the cost of integration and the likelihood of costly errors.

The Common Model Library (CML) defines common expressions of, and the semantics for, concepts that enable information exchange between both management tools and managed resources. These common semantics will enable the integration of tools that span both IT lifecycle phases and management disciplines. Enabling this integration not only reduces the need for human intervention but also enables automation, allowing enterprises to more effectively use their IT resources.

Several published standards exist today which provide integration of management infrastructure, such as the Distributed Management Task Force (DMTF) Common Information Model (CIM) [1], the Internet Engineering Task Force (IETF) Simple Network Management Protocol Management Information Bases (SNMP MIBs) [2], and the TeleManagement Forum (TMF) Multi-Technology Network Management (TMF608) [3]. However, these published standards were scoped to the operational system. They do not have a mechanism to assert requirements throughout broader lifecycle phases (e.g., application lifecycle). To manage an environment which spans the necessary lifecycle phases and management disciplines, a model must allow assertions to be made about the managed environment. Several IT management practices exist, including Information Technology Infrastructure Library (ITIL™) and Control Objectives for Information and related Technology (COBIT™). Where appropriate, CML will leverage existing management data sources, existing industry standards and recognized practice efforts. In particular, CML is not intended to supersede CIM, or any of the other models, but the intention is to have CML work side by side with the other models as seamlessly as possible.

CML will provide an extensible library of models and common modeling elements, as well as rules and guidelines for how models and modeling elements can be extended, combined to form new models, used, abstracted, and subjected to assertions. At the core of the models is a set of constructs for the commonly understood entities that may exist in the managed environment, their states, and their relationships. In order to span lifecycle phases and support the various management disciplines, the models define semantics to capture management information with respect to intent,

including desired state, and time such as historical, current, and future state. To further support management related communication CML will provide a consistent mechanism for capturing assertions and constraints.

CML will be expressed in the Service Modeling Language (SML) and we expect that the SML Interchange Format (SML-IF) will be one of the methods used to exchange documents. CML will include guidelines and best practices for encoding models in SML. The resulting models are intended to be shared, extended, and reused in communications among management tools and managed resources across management disciplines and lifecycle phases.

The first set of goals for the CML Working Group includes creating modeling constructs which are the building blocks for CML. We recognize that SML alone is not sufficient to capture semantic definitions. Semantic definition languages exist (e.g., RDF, OWL) and the group will have to determine an appropriate language to meet the requirements of CML. The use of CML models to facilitate interoperability may require additional work from the CML Working Group. Some issues that arise in using CML models may be provided as recommendations in the documentation, including the need to transmit models in an interoperable manner via an agreed upon set of wire protocols, security considerations including secure access to models and model fragments, the ability to discover models, and to form queries about the models and modeling fragments available from a management information provider.

## **2. Scenarios**

The CML effort is considering a broad spectrum of scenarios in sufficient detail to understand the information that must be exchanged among management tools and managed resources. The following are high-level scenarios that are presented for the purpose of illustrating typical problem areas and how they will be addressed by CML. The set of scenarios below is intended to be a starting point, not exhaustive.

### **2.1 Impact Analysis**

When an IT situation occurs (characterized by an IT outage, performance degradation, or failure to meet service level agreements) IT enterprises need to know the business impact of the situation in order to prioritize efforts to return the IT system to a desired state. Supporting this level of impact analysis requires cross-domain information about how business logic is supported by the services deployed into a service hosting environment. The analysis also requires information about how each deployed service is supported by the resources of its hosting environment. This cross-domain information may be maintained by several different management tools, each of which must exchange information in order to contribute to the overall impact analysis. The use of CML is intended to provide a description for this exchanged information. When an issue occurs that can be predicted to cause one or more services to fail or degrade below agreed upon service levels, the impacted services and the related business logic can be identified by impact analysis, enabling administrators to respond appropriately to the situation.

### **2.2 What-If Analysis**

IT departments often must change their production IT systems without sufficient information to analyze the impact of proposed changes. The same information that is used for impact analysis can be used for "what-if" analysis for these proposed changes. The cross-domain information relating business logic to services, and deployed services to hosting environment resources, combined with information on the capabilities of resources (e.g., capacity, current load, performance characteristics) and the requirements of a proposed change, can be used to check if

the proposed change would cause any existing service deployments to be degraded below their agreed upon service levels. This analysis makes it possible for changes to be validated prior to making any actual updates to the service hosting environment, allowing different approaches to be evaluated. Multiple management tools may be required to participate in the what-if analysis. The use of CML is intended to provide a consistent description of the information to be exchanged between these management tools.

## **2.3 Service Deployment**

An IT department wishes to deploy a business service. This service is composed of business logic implemented as several application components and a set of databases on which the business service depends.

A CML model is used to describe the intended service deployment and to encode the required data base dependencies. The service author will be able to specify security and quality of service requirements, which could be added to or refined during the deployment process, based on local IT policies and business needs. Deployment constraints, such as co-location of selected components, could be similarly specified. The model incorporates the application components by referring to deployment information captured during the development phase. This development-time information is associated with the application throughout its lifetime. An example of this is J2EE™ hosting information captured in J2EE™ EAR files by existing tools.

The service's deployment model must then be mapped to the service hosting environment which is also described in a CML model. To deploy the service, a deployment architect must determine whether or not a suitable hosting environment exists. There will be two cases for each requirement: either there is existing infrastructure that meets the requirement, or to satisfy the requirement infrastructure must be instantiated from a template. For example a template expressed using CML could represent the best way to configure a high throughput application server cluster with single sign-on support while still adhering to local IT policy. Next, the deployment architect adds the target resources and desired configuration that satisfy the deployment requirements to the CML model. This model can be used to validate planned configuration changes and is used to drive resource configuration, and/or installation of new resources in the target data center to satisfy the service's needs.

Implicit in the mapping process are the abilities to evaluate the feasibility of candidate configurations at each stage of refinement for feasibility and to assess how well each is likely to meet the service's aggregate constraints and requirements. Given a consistent way to model and describe requirements, constraints, and capabilities, various degrees of automation can be used for the mapping and configuration evaluation processes. This evaluation process is used, for example, to prevent a situation where a data source configuration for an application or application server is specified but the target database described by the data source is not on the specified host.

## **2.4 Configuration Management Database**

IT Enterprises commonly need to exchange, reconcile, and/or correlate data between management tools that serve different roles in their IT infrastructure. An Enterprise wants to adopt a configuration management database (CMDB) to support ITIL™ practices. The current focus is on integrating their existing asset management, service desk, and performance management tools into a CMDB. To manage risk, the

enterprise IT department decides to take a federated approach: a new CMDB will be constructed that provides a consolidated view of all data about every resource listed in one or more of these management systems without actually replicating the bulk of their contents. Thus, it must be possible to exchange data records with each of these systems, and it must be possible to recognize and associate different data records when more than one system contains information about the same resource. CML could enable this by providing a common taxonomy and common content that each of the management systems includes, along with its own model. The use of CML could enable the reconciliation and/or correlation of data across systems.

### **3. Business Goals**

- Reduce TCO by enabling better integration between the lifecycle phases of managed artifacts.
- Reduce operational expenses and enable greater automation of IT management processes by providing a consistent mechanism for expressing key concepts.
- Reduce errors caused by translating between management data models.
- Enable an ecosystem of management tools that can inter-operate in a heterogeneous environment.
- Enable incremental and non-disruptive adoption by leveraging existing standard management data sources and models.
- Decrease time to market for creating integrated solutions.

### **4. Model Library Overview**

The scenarios in Section 2 demonstrate how CML can enable functionality for a variety of management disciplines. These management disciplines include service desk, configuration management, systems management, network management, change management, service level agreement (SLA) management, asset management, and more. The scenarios also demonstrate how CML must facilitate information sharing across a typical set of lifecycle phases which include planning, development, deployment, and operations phases.

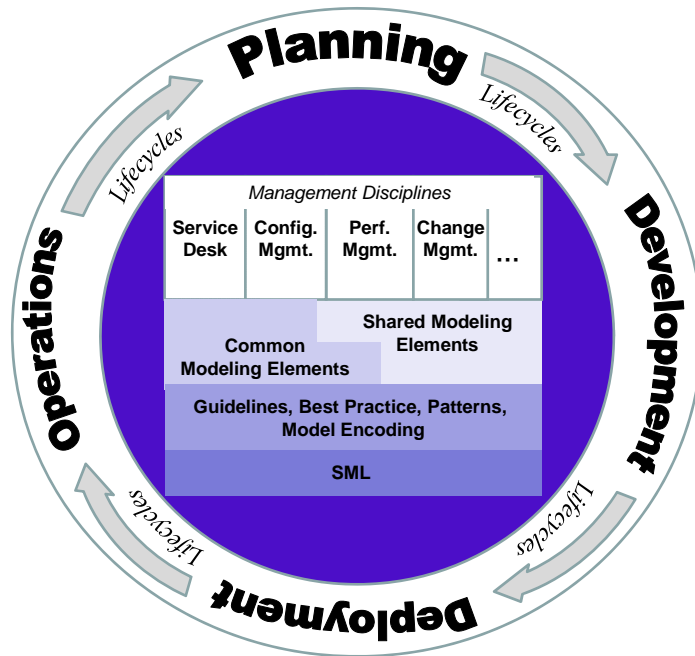
CML will enable IT Enterprises to exchange information between management tools and managed resources that serve different roles in support of the various management disciplines. At its core, CML is a collection of models which are expressed as XML documents that describe IT entities and their relationships. As the basis for common modeling elements and semantics, the models describe information which can be exchanged between management tools and managed resources. This information is about IT systems and includes infrastructure (e.g., servers, application servers, Web services), logical entities (e.g., software license, incident reports, IT roles), and relationships (e.g., hosted, is hosted by, supplied by). Instances of CML models can represent snapshots of real systems, including historical and current states. They can also represent baselines for desired or intended states of the system. In CML, this is accomplished by expressing assertions, requirements, constraints, and goals about the configuration, performance, or lifecycle of these IT entities.

However, CML does not attempt to present a single model or single set of models which will ensure integration. Instead, CML provides a means for creating new models or extending, combining, or evolving existing models. CML uses a single language, in a consistent fashion, to represent common elements that are

communicated between management tools and managed resources, to increase the quality of integrated management tools and managed resources. In addition, the use of CML is expected to reduce the time necessary to perform the integration.

Figure 1 (below) depicts CML and its relationship to the variety of management disciplines and lifecycle phases.

**Figure 1. CML Overview**



## 4.1 Model Library

### 4.1.1 Common Language

CML uses SML as a foundational language for encoding models. SML is based on XML Schema and adds references between documents. SML uses Schematron [4,5] for constraints. CML models are expressed in SML and are often communicated, or interchanged, in SML-IF. For more information on SML and SML-IF see Section 5 below.

### 4.1.2 Consistent Information Encoding

CML dictates the consistent usage of SML by constraining the ways in which information is encoded. However, requiring SML as a modeling language isn't sufficient to ensure that models will be encoded consistently. In fact, if the industry were to just use SML, there would soon be many proprietary model dialects, and management integration would continue to be time-consuming. Using a wide set of usage scenarios, the CML effort will develop guidelines, best practices, patterns and model encodings. These will guide the development of the Common and Shared Modeling Elements described in the next two sections.

### 4.1.3 Common Elements

Management tools communicate using common elements such as events, properties, time series, requirements, constraints, capabilities, relationships, etc. In CML, these

common modeling elements are elements that are re-used across most, if not all, management disciplines.

#### **4.1.4 Shared Elements**

CML shared modeling elements are common in management and are re-used across a subset of management disciplines but are not necessarily widely applicable. Examples of shared modeling elements are incidents, configuration items, service level agreements and objectives. Shared modeling elements are important in product-to-product integration scenarios; the use of incidents in the integration of monitoring applications with service desk applications is an example.

#### **4.1.5 Semantic Definitions**

Common semantics can be applied to further bring down the cost of integration. Even with CML, there will be lots of different models and information. Semantics can be used to describe information and capabilities to facilitate searching and the reconciliation of information. Standardized vocabularies and semantic information can be associated with models to further facilitate integration.

### **4.2 Additional Design Principles and Goals**

- CML will adopt a scenario driven approach for creating guidelines, best practices, and modeling elements.
- CML models will facilitate interoperability between the tools and infrastructure that implement the various management disciplines.
- CML models are intended to be self-documenting.
- CML will borrow from and build upon existing standards.
- CML will be extensible by multiple parties independently crafting their own models, without losing the core concepts described within the library.
- CML models should be modular, i.e., richly coupled internally, and define relatively independent references between them.
- With CML, it will be possible to express and relate different aspects of entities.
- CML will enable building schemas from modeling elements and will separate concepts in a manner that enables reuse without requiring the restructuring of existing models or CML.
- CML will provide mechanisms to enable forward and backwards compatibility between iterations of a model as the various implementations and the standard evolve.
- CML will enable the expression of time, intent, and lifecycle phases for modeled entities.
- CML will strike a balance between unification into a single common model and the use of semantics as a mechanism to glue multiple disparate models together.

### **4.3 Non-Goals**

- CML does not address implementation. CML is used to describe various management domains and best practices, and therefore describes states.

However, CML does not address how implementations reach particular states or meet goals with respect to states.

- CML is not designed for use as an internal data representation. Typically, internal data formats are optimized for their specific applications, which is outside the scope of CML.

## 5. CML and SML

CML models are expressed using Service Modeling Language (SML) and exchanged using SML Interchange Format (SML-IF). SML is well suited for encoding CML models because it defines an XML representation for modeling concepts like references between coherent sets of documents and XML Schema extensions to constrain the references. SML also, through the use of Schematron, provides the ability to apply constraints to a model.

A brief description of SML and SML-IF are:

- SML is an XML Schema-based language providing constructs for modeling the elements in complex IT services and systems, their relationships and constraints
- SML uses Schematron for defining constraints. Schematron is an ISO/IEC standard
- SML Interchange Format (SML-IF) defines an XML serialization format for SML-based models
  - It defines a container XML document for all definitional and/or instance documents of a model
  - It is designed to facilitate the exchange of SML model data, preserving content and relationship data

SML and SML-IF were submitted to the W3C by the CML member organizations and was accepted in March 2007 by the W3C .The specifications are now available at W3C's web site:

<http://www.w3.org/XML/SML/> (workgroup site)

## 6. Deliverables

The work of the CML effort is expected to include:

- A library of models expressed as SML documents
- Common and shared modeling elements expressed as SML document fragments
- Guidelines for encoding models
- Patterns & best practices
- Semantic definitions
- Examples and scenarios
- Compliance or conformance suites, validated at group workshop meetings

## 7. Future and Next Steps

The work of the CML effort is ultimately intended to produce specifications for submission to an industry standards group for standardization.

The immediate next set of deliverables from the CML Working Group is:

- Additional scenarios, at a high level of detail, which describe the use of CML. The scenarios in this document will be defined to the same level.
- The initial set of CML model encoding guidelines.

## 8. References

[1] Common Information Model: (<http://www.dmtf.org/standards/cim/>)

[2] Simple Network Management Protocol RFCs document numbers 3411 through 3418: (<http://www.ietf.org/rfc.html>)

[3] TeleManagement Forum TMF608 : (<http://www.tmforum.org/page32957.aspx>)

[4] Document Schema Definition Language (DSDL) – Part 3: Rule-based validation – Schematron:

([http://standards.iso.org/ittf/PubliclyAvailableStandards/c040833\\_ISO\\_IEC\\_19757-3\\_2006\(E\).zip](http://standards.iso.org/ittf/PubliclyAvailableStandards/c040833_ISO_IEC_19757-3_2006(E).zip))

[5] An Introduction to Schematron:

(<http://www.xml.com/pub/a/2003/11/12/schematron.html>)