

White Paper
CMII 855B

**Configuration Items:
History and Current Usage**



Prepared by
CMII Research Institute

White Paper CMII-855B
Configuration Items: History and Current Usage

Revision Record

<i>Revision</i>	A	B	
<i>Released by</i>	WWG	WWG	
<i>Release date</i>	05/22/08	12/29/09	
<i>Authority</i>	062-WP	063-WP	
Page 1	A	B	
Page 2	A	B	
Page 3	A	B	
Page 4	A	B	
Page 5	A	B	
Page 6	A	B	
Page 7	A	B	
Page 8	A	B	
Page 9		B	

Purpose of This White Paper

To describe how configuration items (CIs) were originally used in the defense environment, their current use in other environments, and better ways to achieve the same objectives.

OUTLINE

- **Configuration Items per the MIL-STDs**
- **MIL-STD Criteria for Selecting CIs**
- **Other Related Definitions from MIL-STD-480**
- **CIs per the Replacement Standard, EIA-649A**
- **ITIL and SEI's Criteria for Selecting CIs**
- **Summary — From a CMII Perspective**
- **Conclusions**
- **Recommendations**

Configuration Items per the MIL-STDs

Configuration items (CIs) were first used by the Department of Defense in the late 1960s, early 1970s. The most explicit instructions on how they were to be used, and how they relate to other configuration management practices, were provided in the following Military Standards:

*MIL-STD-483A
Configuration Management
Practices for Systems, Equipment,
Munitions and Computer Programs
4 June 1985*

*MIL-STD-480B
Configuration Control —
Engineering Changes,
Deviations and Waivers
15 July 1988*

The use of CIs in the defense environment was very similar to how models are used in commercial environments. The following excerpts from the above standards describe how CIs were used.

Part Number

All discrete parts, assemblies and units shall be identified by part numbers in accordance with DOD-STD-100.

MIL-STD-483A, Appendix IX

Configuration Item Identification Numbers

A family of like-units of a configuration item that individually satisfies prescribed functional requirements shall (in addition to a part number) be identified by an unchanging base number such as a CI identification number or a type-model-series designator. The CI number

- a. shall establish a base for serializing individual units of a CI;*
- b. shall not change when the unit is modified;*
- c. shall remain the same even though the CI may have more than one application or be reprocedured from different contractors;*
- d. shall be composed of seven digits of alpha-numeric characters.*

MIL-STD-483A, Appendix IX

Serial and Lot Numbers

A single unit or lot in a family of like-units of a CI shall be permanently and uniquely identified by a serial or lot number. The serial or lot number shall not exceed 15 digits of alpha-numeric characters. It shall be assigned sequentially and shall be numeric in the last four digits.

MIL-STD-483A, Appendix IX

MIL-STD Criteria For Selecting CIs

Configuration Item (CI)

An aggregation of hardware, firmware, software or any of its discrete portions, which satisfies an end use function and is designated for configuration management. Any item required for logistic support and designated for separate procurement is a CI.

MIL-STD-480B page 6

Level of Government Control

The CI selection process separates the elements of a system into individually identified subsets for the purpose of managing their development. CI selection reflects an optimum management level during acquisition. The contracting agency contracts at this level.

MIL-STD-483A page 13

Logistics and Life Cycle Considerations

The selection of CIs should be determined by the need to control a CIs inherent characteristics or its interface with other CIs. Selection is normally accomplished through the systems engineering process in conjunction with configuration management and the participation of logistics. Choosing too many CIs increases the cost of control. Choosing too few runs the risk of too little control. CI selections are made accordingly.

MIL-STD-483A, pages 113 and 114

Common Subsystem Considerations

On development programs that are common to more than one system, the basic CI should be that which is common to all applications.

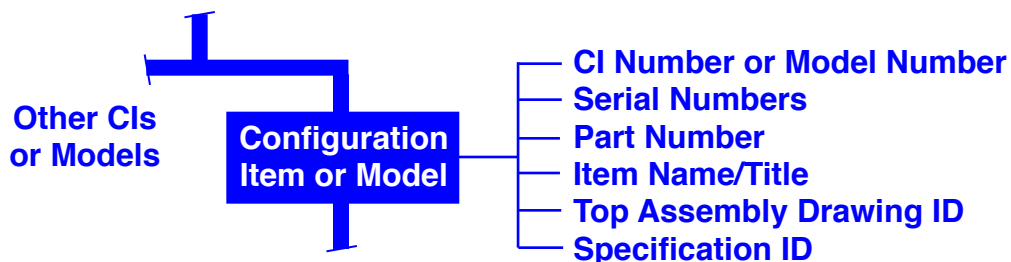
MIL-STD-483A, page 114

Interfaces

Interfaces among CIs should be simple. Functions that are highly interdependent should be allocated to the same CI.

MIL-STD-483A, page 115

Government oversight and the fully coordinated Engineering Change Proposal (ECP) process was required for items designated as CIs.



Other Related Definitions From MIL-STD-480

Baseline

Baselines, plus approved changes, constitute the current approved configuration. For CM purposes, there are three baselines which are established sequentially, as follows:

- *Functional Baseline (high-level system)*
- *Allocated Baseline (separate developmental segments or CIs)*
- *Product Baseline (as produced)*

Configuration Identification

The selection of documents to comprise the baseline for the system and CIs and the numbers and other identifiers affixed to items and documents.

Configuration Control

The systematic proposal, justification, evaluation, coordination, approval or disapproval of proposed changes in the configuration of a CI after formal establishment of its baseline.

Configuration Management

A discipline applying technical and administrative direction and surveillance to (a) identify and document the physical and functional characteristics of CIs, (b) audit CIs to verify conformance to documented requirements, (c) control changes to CIs and their related documentation and (d) record and report the status of proposed changes and the implementation status of approved changes.

CI's Per the Replacement Standard, EIA-649A

The EIA-649A "National Consensus Standard for CM," which replaced the military standards for CM, includes provisions for serializing individual like-units but there are no provisions for CIs or model numbers.

After the 1994 cancellation of the military standards, government procurement agencies now specify only the requirements. Contractors are now fully responsible for CM and design control. Since government oversight was eliminated, CIs were eliminated. However, an unchanging base number against which to serialize is still needed.

ITIL and SEI's Criteria for Selecting CIs

IT systems and software developers define CIs differently:

"A configuration item (CI) is any component of an IT infrastructure, including a documentary item such as a service level agreement or a request for change, which is (or is to be) under the control of configuration management and therefore subject to formal change control. The lowest level CI is normally the smallest unit that will be changed independently of other components. CIs may vary widely in complexity, size and type, from an entire service (including all its hardware, software, documentation, etc.) to a single program module or a minor hardware component. All existing or potential service problems will be capable of being linked to one or more CIs"

ITIL: Information Technology Infrastructure Library

http://www.knowledgetransfer.net/dictionary/ITIL/en/Configuration_Item.htm

Guidelines for selecting CIs are provided in Software Engineering Institute's (SEI's) Capability Maturity Model Integration (CMMI) as follow:

- *Work products that may be used by 2 or more groups;*
- *Work products that are expected to change over time;*
- *Work products wherein a change to one causes others to change;*
- *Work products that are critical for the project.*

The following work products are examples of CIs per CMMI:

- *Process descriptions;*
- *Requirements;*
- *Design;*
- *Test plans and procedures;*
- *Test results;*
- *Interface descriptions;*
- *Code/Module*
- *Tools (e.g., compilers)*

Per CMMI, each CI is assigned a unique identifier and its important characteristics are specified (such as):

- *Author;*
- *Document or file type;*
- *Programming language.*

In other words, each element within an IT system/software development project that needs to be formally controlled is designated as a CI.

Summary — From a CMII Perspective

CM evolved in the defense environment and the original application of baselines, like the original scope of CM, was limited to design definition. The CM policy was that only items designated as CIs would be placed under CM — which meant joint control by the government.

CIs, as used in the defense environment prior to cancellation of the military standards, are equivalent to models as used commercially. Both CIs and models provide a base number against which to serialize each unit or lot within a like-family of units or lots.

The defense environment took it a huge step further and made CIs the basis for making government oversight an integral part of the design baseline and change control process. The oversight requirement went away when the military standards and their "how-to directives" were replaced with performance-based specifications. This made it appear that CIs were no longer needed.

Developers of commercial IT systems and software concluded that the CI concept still had merit. They concluded that work products which may impact quality, schedule or cost should be formally placed under CM control. This objective is being achieved by assigning a CI designation to such work products.

Conclusions

Whether called a CI or a model, an unchanging base number against which to serialize units within a like-family of units is still needed. The military standards, when cancelled, were to be replaced with best commercial practices. Commercial environments use models. CIs should have been replaced with models.

Placing work products that could impact quality, schedule or cost under formal CM control was a good decision. This can be achieved by designating each work product as a CI, or by placing each work product in a baseline. Any work product that resides in a baseline is, by definition, under formal CM control. From a CMII perspective, the baseline approach is a better solution.

Recommendations

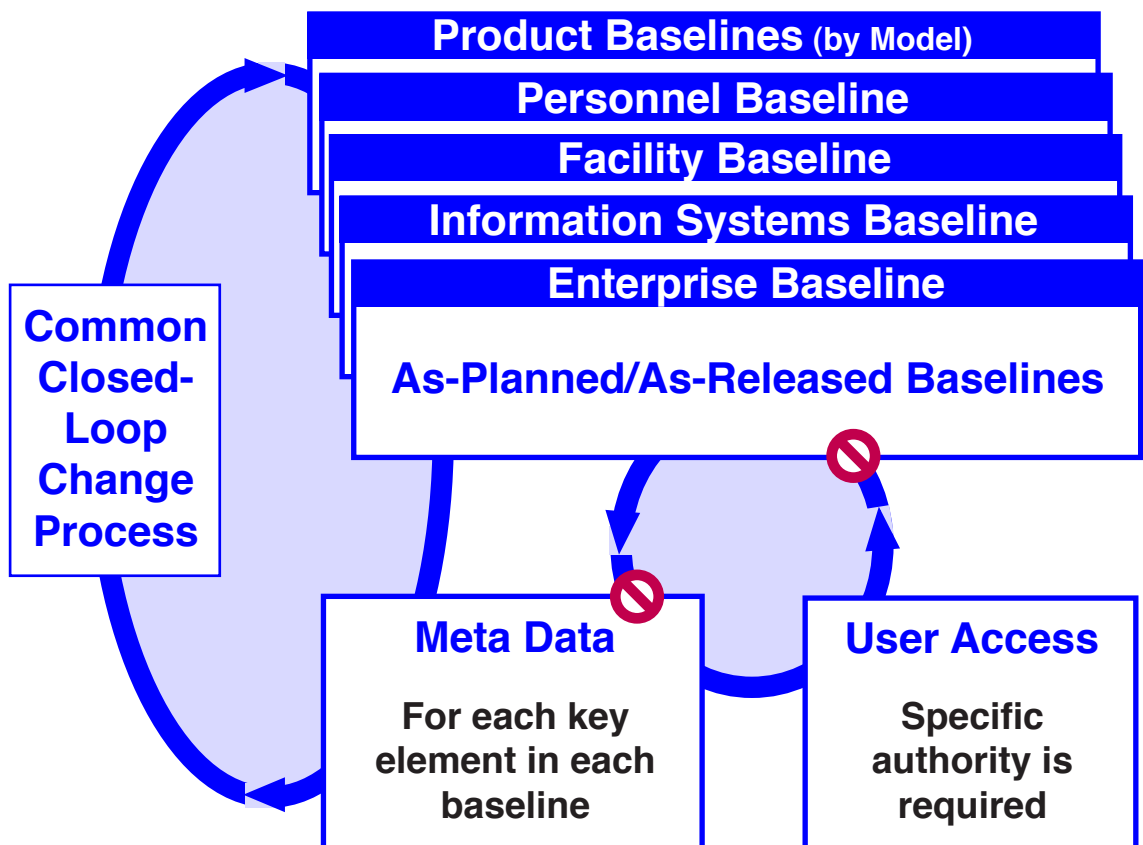
All work products should reside in a properly structured and integrated set of baselines. Criteria for selecting those work products should be expanded to include anything that could impact safety, security, quality, schedule, cost, profit and the environment.

As-Planned/As-Released Baselines and Meta Data

Appropriate baselines are called as-planned/as-released baselines and include visibility of planned changes. They ratchet forward with each change (or Enterprise Change Notice) and are always current.

Appropriate types of as-planned/as-released baselines are shown below. Each key element in each baseline has its own meta data.

There is a template for each type of baseline and each type of meta data. Meta data is used to identify and manage the linkages between the key elements in each baseline.



Template for an End-Item Product Baseline

The example below represents the template for an as-planned/as-released baseline for an end-item product. In this example, the identity of the baseline is also the model or name for the product.

The baseline for a physical entity such as a product, system or facility includes its entire hierarchy of physical items. Each item at each level is linked to its own unique set of documented requirements.

Planned changes are included in the baseline. Each ECN has an effectivity. Each document has a release date and an effective date. Shading is used to highlight the superseding items and/or documents.

AS-PLANNED/AS-RELEASED BASELINE											
Model/Name _____						Date/Time ____/____					
Physical Items			Documented Requirements						Changes		
Hierarchy (Qty per)	Item ID No.	Item Name	Type	Number	Rev	Rel. Date	Eff. Date	ECN Auth.	A/D	Effec- tivity	ECN Auth.
0	NNNN	XXXXXX	TT	NNNN	R	DMY		---			
1	9876	End Item	TT	NNNN	R	DMY	DMY	NNN			
			TT	NNNN	R	DMY	DMY	NNN			
			TT	NNNN	R	DMY	DMY	NNN			
			BM	9876	A	DMY	DMY	NNN	D	DMY	NNN
			BM	9876	B	DMY	DMY		A	DMY	NNN
1	NNNN	XXXXXX	TT	NNNN	R	DMY	DMY	NNN			
1	1234	XXXXXX	TT	NNNN	R	DMY	DMY	NNN	D		NNN
1	2345	XXXXXX	TT	NNNN	R	DMY	DMY		A		NNN

Foundation for an Efficient Change Process

Baselines are closely coupled with a closed-loop change process. A change process cannot be fast and efficient if the information being changed is not properly identified, structured, linked and owned. These prerequisites are best satisfied with properly structured and integrated baselines and meta data. Together, they provide the foundation for a fast and efficient change process.