



## **COMPUTER-MANAGED INSTRUCTION**

### **AICC CMI SUBCOMMITTEE**

#### **SCOPE**

This document recommends guidelines that promote the interoperability of CMI systems. Interoperability means the ability of a given CMI system to manage CBT lessons from different origins. It also means the ability for a given CBT lesson to exchange data with different CMI systems.

#### **RECOMMENDATIONS**

For MS-DOS and MS-Windows courseware, the AICC recommends a(n):

- CBT delivery system that uses AICC-compatible CMI-to-lesson communication files and rules.
- CMI system that uses AICC-compatible CMI-to-lesson communication files and rules.
- CMI system that is able to export and import AICC-compatible course structure files.
- CBT delivery system that generates AICC-compatible lesson evaluation files.
- Authoring tool that is able to create lessons that generate AICC-compatible data files.

AICC-compatible files, rules, and compliance tests are defined in the documents identified below. An AICC-sponsored Independent Testing Laboratory can verify compliance of CBT courseware and CMI systems to this AGR

#### **RATIONALE**

In the past, authoring systems made the CBT customer a captive of the authoring system vendor. If the customer wanted to take advantage of CMI features in his courses, he had two choices:

#### **Caveats...**

The data contained in this document has been collected by the AICC as an information resource of computer-based training programs. Neither the AICC nor any of its members assumes nor shall any of them have any responsibility for any use by anyone for any purpose of this document or of the data which it contains.

- Design his own CMI system with his authoring system tools, or
- Purchase a CMI system from the same vendor who supplied the authoring system.

In either case, the resulting CMI system works only for a single vendor's CBT lessons. This is fine, until the customer acquires CBT courseware designed with a different authoring system. This AGR recommends guidelines that promote the interoperability of CMI systems.

## REFERENCE DOCUMENTS

The AICC CMI functional specifications and requirements are described in a single document, *CMI Guidelines for Interoperability* (AICC document CMI-001, approximately 250 pages).

*AICC/CMI Certification Testing Procedures* (CMI-003, approximately 100 pages) defines the procedures and criteria for compliance testing of CMI systems/CBT Courseware to the specifications in the CMI-001 document.

The latest version of these two documents can be found at [www.aicc.org](http://www.aicc.org).

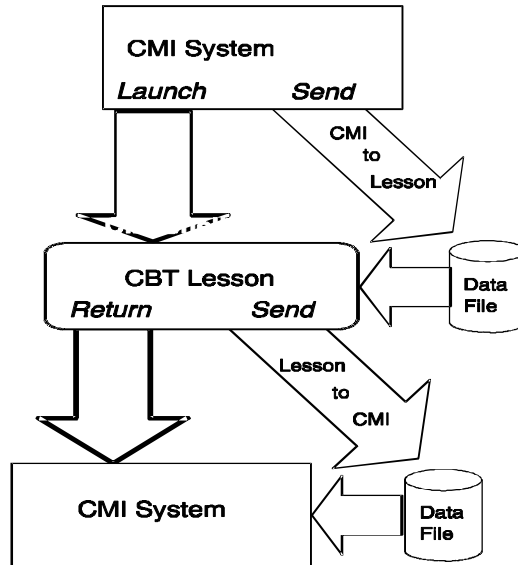
## OVERVIEW OF CMI GUIDELINES FOR INTEROPERABILITY

### CMI Management of CBT

There are two aspects of the AICC approach to enabling interoperability of CMI systems with different CBT systems.

- 1) **Lesson launch:** The CMI should have a standard approach to CBT lesson initiation, and
- 2) **Communication:** The CMI should have a standard approach to providing information to the CBT lessons, and receiving information from the CBT lessons. AICC Guidelines define two files to enable this communication:

<b>Data File: CMI to CBT</b>	Lesson start-up information. Information that the lesson may need about the student.
<b>Data File: CBT to CMI</b>	Information required by the CMI system to record student performance and perform the next lesson routing or assignment.



**Figure 1: CMI Management of CBT**

This is how the interoperability works:

- The CMI system creates a file containing the data necessary to start-up a CBT lesson. The file is created just prior to the initiation of the CBT system.
- Once the CBT lesson is initiated, it reads the data file created by the CMI system and then deletes it.
- The CBT system must create a file containing data to be passed back to CMI so that the CMI system can update its student performance data and make the next assignment (perform routing activity).
- The CMI system passes back the file name for the lesson-to-CMI data file as part of the CMI-to-lesson core data.
- When the student leaves the lesson, the CBT system updates and completes the file of information for the CMI system.
- The CMI system reads the CBT-to-CMI file, updates applicable student data, and determines the next student assignment or routing activity.

CMI and CBT Lesson communication is two way. The CMI system sends information to the lesson when it begins. The lesson sends information to the CMI system when the lesson ends. Information is sent in a file -- two files actually. The first file is created by the CMI system, and the second is created by the lesson.

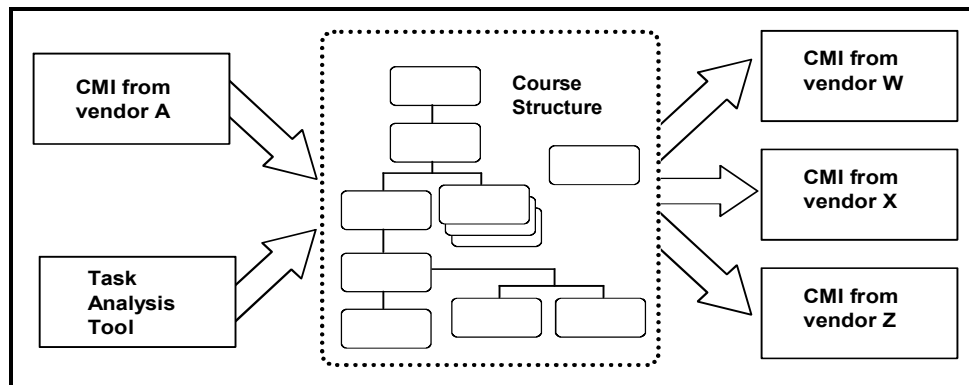
The CMI to CBT File is information that a typical lesson obtains from a CMI system to enable it to perform the functions expected of it. The file consists of Core items and optional items. Core items must always be provided by the CMI system to be AICC compliant. Lessons may always depend upon core items being available. Optional items are group and keyword data which may be needed by a lesson to perform optimally.

The CBT Lesson to CMI File is information that a lesson must/may make available to a CMI system.

**Course Structure Interchange**

A course may be as simple as a few lessons to be viewed sequentially, or it may be as complex as hundreds of lessons, some of which are prerequisites to others and some of which may be experienced in any order. Basically, courses have two components: instructional elements and structure.

The instructional elements are all the lessons, tests, and other assignable units (AUs) in the course. Frequently, the content elements also include all of the objectives to be mastered in the course. In defining a structure, the developer frequently groups lessons for assignment. In other cases the designer defines complex lesson hierarchies.



**Figure 2: Course Structure Interchange**

The structure determines the order in which these are to be experienced by each student. The part of the CMI system that sequences the course content, is referred to as the *router*.

There are at least two circumstances in which guidelines for moving courses from one environment to another are useful. The first assumes a course is complete and is being transferred from a vendor or manufacturer to an airline -- moving from one CMI system to another. The second assumes a course is being designed in a tool other than a CMI system -- moving course design into CMI. Having a standardized mechanism for describing course content and structure, enables CMI systems to "ingest" a new course with minimal manual effort.

The AICC has identified seven files (some optional) that can be used to describe a course's content and structure. Additionally, the AICC guidelines define five levels of complexity in describing the course structure. Increasing the level of complexity should result in:

- Less effort to review and modify the CMI system after importing the data.
- More complete description of the designer's intended usage of the course material.

The level of complexity determines the number of files required and the amount of information required in each file. The following table briefly describes the contents or purpose of each file.

<b>Course Description File</b>	Information about the course as a whole including a textual description of the course, and general makeup of the course -- the number and type of elements.
<b>Assignable Unit Table</b>	Information about the assignable units (AUs) in the course. Each AU has its own record (or row in the table). The information includes the name of the AU, its ID, and the mastery score for that AU.
<b>Descriptor Table</b>	A complete list of every course element in the course including: AUs, Blocks, Objectives, Complex Objectives. It is used as the basic cross reference file showing the correspondence of system-generated IDs with user-defined IDs for every element.
<b>Course Structure Table</b>	The basic data on the structure of the course including all of the AUs and blocks in the course, showing how they are organized. Finally, it implies the order in which these should be taken.
<b>Objectives Relationships File</b>	Objectives have complex and variable relationships to other elements of a course. This file defines all of these relationships. This file is optional, depending on the level of the course description.
<b>Prerequisite Listing</b>	Sometimes it may be desirable to prevent a student from entering a lesson until he has met certain prerequisites. This file allows that sort of constraint to be placed on each block or AU in a course. There are three levels of complexity that may be used in describing prerequisites: <ul style="list-style-type: none"> <li>• a single prerequisite AU or block to be defined for each element in the course</li> <li>• prerequisites to be defined in the form of a logic statement (with "ands" and "ors")</li> <li>• the definition of prerequisites for each mode (Review, Browse, Normal) for the lesson</li> </ul>
<b>Completion Requirements</b>	While lesson and objective status is determined within the lesson by the logic designed into it, this is not true of blocks. Blocks are created specifically to describe a course structure. Similarly Complex Objectives are defined in terms of other structure elements. Therefore, block and complex objective status must be determined by the CMI system. The Completion Requirements file is designed to allow the explicit specification of when a block or objective is complete when it does not conform to the defaults for completion. It is essentially an exception file.

### Storing Lesson Evaluation Data

Lesson evaluation data includes information that a CBT lesson or test generates on the behavior of a student. It may include such items as a student's responses, latency, and path through a lesson. Standardizing the format of the student records permits multiple tools to use the information.

Lesson evaluation data is contained in several files. File names for this data are passed to the lesson from the CMI system. If the file already exists, the lesson appends the data. If the file does not exist, the file is created and the data deposited.

The analysis of the information is not the subject of these guidelines. What is covered here is essentially raw data. These files are all optional. Up to five files may be required to store all of the information desired from a CBT lesson. The files are the following:

<b>Comments File</b>	A journal file that contains freeform feedback from the student
<b>Interactions File</b>	An interaction is a recognized and recordable input or group of inputs from the student to the computer. These interaction records will commonly be student responses to questions.
<b>Objectives Status</b>	Comprehensive information on objectives, including their ID and their status (passed, failed, or not attempted.)
<b>Path File</b>	This file allows an analysis of what path the student took through a lesson.