GUIDELINES FOR CMI INTEROPERABILITY: THE AVIATION INDUSTRY STEPS FORWARD

Jack Q. Hyde
Boeing Commercial Airplane Group
Seattle, Washington

Anne Montgomery
Independent Computer Systems Consultant
Las Cruces, New Mexico

ABSTRACT

One of the most recent actions of the Aviation Industry CBT (Computer-Based Training) Committee (AICC) was to publish guidelines for the interoperability of Computer Managed Instruction. This paper describes the AICC guidelines for interoperability of CMI systems. It addresses

- How CMI systems in general function
- The value of interoperability
- Achieving interoperability: An overview of guidelines in three areas
  1) CMI/CBT interoperability: How different CMI and CBT systems from different vendors can work together.
  2) CMI/CMI interoperability: How different CMI systems can pass course structure and student management rules to other CMI systems.
  3) Lesson evaluation tools: How different data analysis tools can work with CBT from different vendors.

ABOUT THE AUTHORS

Jack Hyde has been working in Flight and Maintenance Training at Boeing for over 20 years. He started as a classroom instructor and training developer in the flight training ground school. In 1977, he designed and implemented his first CBT lesson using the PLATO system. Since then he as worked with WICAT and Authorware CBT systems as well. Currently he is working in a group called Customer Training Technology with a job title of Computer Technology Analyst.

Ms. Montgomery has over 20 years experience in the design, development, and implementation of Computer-Based Training (CBT) Systems, Computer-Managed Instructional (CMI) Systems, and Training Management Systems (TMS). She has developed customized CMI systems for both commercial and military training applications. In her role as technical coordinator of the Aviation Industry CBT Committee, she has worked with airframe manufacturers, CBT vendors, airlines, and CBT developers in the definition of CMI interoperability guidelines. Ms Montgomery is an independent consultant residing in Las Cruces, New Mexico.

GUIDELINES FOR CMI INTEROPERABILITY: THE AVIATION INDUSTRY STEPS FORWARD

Jack Q. Hyde
Boeing Commercial Airplane Group
Seattle, Washington

Anne Montgomery
Independent Computer Systems Consultant
Las Cruces, New Mexico

THE AICC

The Aviation Industry CBT (Computer-Based Training) Committee (AICC) is a 5-year old consortium of international CBT professionals. Membership includes the major airframe manufacturers and their suppliers, leading aviation-industry CBT vendors, airlines, and other standards-making and regulatory agencies. Its primary purpose is to generate guidelines which promote the economic and effective use of CBT within the aviation industry. In pursuit of this purpose, the AICC has published guidelines for the purchase of CBT delivery stations, standards for digital audio, and recommendations for selecting an operating system.

After conducting an airline survey to determine needs and desires regarding Computer-Managed Instruction (CMI), the AICC has recently completed the task of creating CMI guidelines. The goal of the guidelines is three-fold:

1) To allow each CMI system to be able to manage a variety of CBT lessons from different vendors, and
2) To enable CBT lessons to operate with a variety of CMI systems and data analysis tools, and
3) To allow course structure and student management rules to be passed from one CMI system to another

CMI OVERVIEW

CMI stands for Computer Managed Instruction. A CMI system is more than a scheduler of CBT materials. CMI systems are capable of managing both online (CBT) and offline instructional activities and tests. In general a CMI system has one or more of the following five components:

1. A component used for the development of course structures.

2. A testing component used for the development and administration of offline and online tests. Testing can be handled via
   - The CMI system
   - A separate test system (off line)
   - Traditional CBT.

   Each of these must be able to report test results to the CMI system.

3. A student rostering component enables entering student names and demographic data. Students may be grouped into classes with this component.

4. A component which provides student assignment management or routing including:
   - Administrator/instructor functions to oversee the day-to-day training operations and intervene when necessary
   - Assignment manager functions to control student assignments based on sets of rules (both predetermined and user-defined)
   - Standard approach to lesson initiation to provide a method for the CMI system to start-up lessons from different CBT vendors
   - Student logon functions to control and manage student access, maintain student-accessible data records, and display the student's current assignment

5. A component which provides student data collection and management including:
   - Functions to collect and maintain performance data on students at all levels of courseware presentation
   - Functions to provide standard analyses and outputs on performance data collected

INTEROPERABILITY OVERVIEW

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

1) Design his own CMI system with his authoring system tools, or
2) 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 aquires CBT courseware designed with a different authoring system, from a different vendor.

Several circumstances can motivate a customer to use CBT courseware incompatible with his CMI system.

- A manufacturer delivers incompatible courseware with a new airplane purchase.
- An airline purchases courseware from a vendor that uses a different authoring system.
- A customer decides to design new CBT lessons with a different authoring system.

There are many reasons a customer may wish to continue to use a single CMI system instead of multiple systems (different CMI systems for different groups of CBT lessons)

- Instructors are already familiar with a CMI system, and training on a new system would take time. This impacts the speed with which new courseware can be used, and the cost of training how to use it.
- It is desirable to maintain the student's overall "look and feel" in the airline's courseware. (The CMI/student interface provides a significant part of a course's look and feel.)
- Maintenence of two different CMI systems is more complex than maintaining a single system.
- The current CMI system has features and functions not available with the CMI associated with the new courseware.
- *There is a desire to add some new lessons designed with a different authoring system to an existing course. A single CMI system is desirable for the entire course.*

This paper describes the three aspects of CMI interoperability covered by the AICC guidelines; and suggests reasons why these aspects of interoperability are desirable.

The three aspects of interoperability discussed are:

- CMI management of CBT lessons.
- Moving course structure between systems.
- Storing lesson evaluation data.

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 and instructions to the CBT lessons, and receiving information from the CBT lessons. The AICC Guidelines define two files to enable this communication:
   - CMI to CBT: Lesson start-up information.
   - CBT to CMI: Information required by the CMI system to record student performance and perform the next lesson routing or assignment

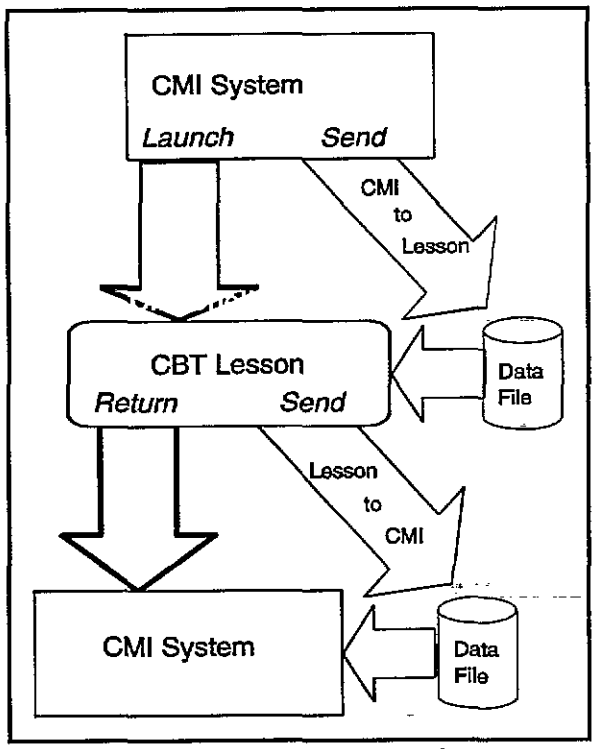

Figure 1: CMI Management of CBT

Process Summary -- Essentially this is how the interoperability works:

1. 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.

   The name of the CMI-to-CBT lesson data file must be known to the CBT application.

2. Once the CBT lesson is initiated, it reads the data file created by the CMI system and then deletes it. (Some lessons may not need this input file simply because student information is not necessary for the lesson.)

3. 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 in the file name for the lesson-to-CMI data file as part of the CMI-to-lesson core data.

4. When the student leaves the lesson, the CBT system updates and completes the file of information for the CMI system.

5. The CMI system reads the CBT-to-CMI data file, and using the information updates applicable student data kept by the CMI system and determines the next student assignment or routing activity.

   It is the responsibility of the CMI system to delete the CBT-to-CMI data file either immediately after determining the student's next assignment/routing activity or in such a manner as to insure that the disk space is managed properly and that there is no leftover data confusing the lesson.

Moving Courses

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 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. The AICC Guidelines accommodate both of these needs with the concept of a block. Blocks are simply groupings of instructional elements or other blocks.

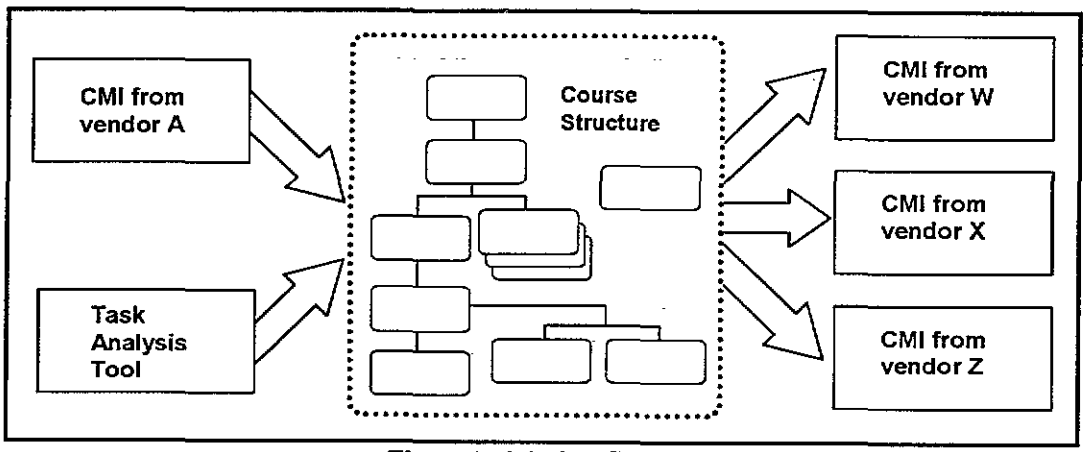

Figure 2: Moving Courses

The structure determines the order in which these are to be experienced by each student. The order may be quite complex, depending on prerequisites, or even student performance. 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.

Transferring a new course into the existing CMI system manually, requires typing hundreds of lesson names, and duplicating all of the sequencing *information. This re*quires a significant number of man hours. Having a standardized mechanism for describing course content and structure, enables CMI systems to "ingest" a new course with minimal manual effort.

There are many tools, other than a CMI system, which may be used to design a new course. One of the most common is a Task Analysis tool. If a course design tool can output a standardized description of a course, the CMI system can pull in the new course from that description. This can save hundreds of man hours of retyping and inputting data.

Storing Lesson Evaluation Data

Lesson evaluation data includes information that a CBT lesson or test generates on the behavior of a student (i.e. his performance. It may include such items as a student's responses, latency, and path through a lesson.

Lesson evaluation data can be used for

- Student performance analysis. Data collection of the student's interaction with the lesson. This helps to determine what the student knows, and what he learns. Comparing individual student progress with his peers gives a measurement of individual rate of learning.
- Item analysis. This can indicate how well an element of instruction trains; or how well a test question measures student performance. This enables quality *control of the testing and instruction.*
- Attitude survey. The determination of how well the student likes the courseware. How well the student feels the courseware is working. This aids in measuring customer satisfaction.
- Path optimization. The determination of the best sequencing of lessons and tests for a specific student. The determination of what material may be skipped by a student. The determination *of what supplementary material or* remediation is required by a student.

Standardizing the format of the student records permits multiple tools to use the information. By having standard interchange

formats, the market for analysis tools becomes much larger than just a single vendor's customers. Vendors are therefore encouraged to create sophisticated, easy-to-use analysis tools because of the payback of a larger customer base.

INTEROPERABILITY KEY: THE FILE

CMI and CBT systems must be able to communicate with each other in order to work together. Communication is essentially a flow of data from one program to another, or from one system to another.

The three data flows required for interoperability discussed in this document are:

- CMI ←→ CBT
- CMI → CMI
- CMI → Lesson-evaluation

In each of these cases the data flow can be handled with files. By creating guidelines for file format and content, the data can be understood by any CMI or CBT system.

The AICC selected two file formats for the data in these flows -- both are ASCII formats that are readable with any simple text editor:

- Microsoft Windows .INI file
- Comma delimited text file.

MS Windows INI Files

This file structure is based on the Microsoft WINDOWS *.INI files. The INI file contains three types of data -- **group, keyword**, and **comment.** The structure of the file and these data types are discussed below.

Groups provide a mechanism for dividing a file into manageable segments that are more easily accessed by data retrieval routines. They also provide a means to organize a file of data into logically related parts. This is helpful for human-processing of a file as well as computer processing.

Groups tend to be large data items, generally several lines in length. A group extends from its group identifier to the next group identifier, and may include multiple lines. Although groups may contain keywords, they may not contain other groups.

Keywords are names of data items that are limited in size to a single line. This generally limits the data to 60 or 70 characters. The data items associated with a keyword are referred to as keyword arguments or keyword values.

Comments are text that is of use to a human viewing a file. They are ignored by a computer processing the data in the file.

Table 1: INI File Elements

Appearance in file	Element name
[group]	Group
keyword=parameter	Valid Keyword
; groups and keywords	Comment
; may have comments	

Example -- This file was created by a Lesson to pass information to a CMI system.

Table 2: Example Windows INI File

```
[CORE]
LESSON_STATUS = Passed
LESSON_LOCATION = End
SCORE = 87
TIME = 00:25:30
; this is the core group of data
; this is the lesson performance data for
; a passed lesson that
; required a time of 25 minutes,
; 30 seconds.
; The student recieved a raw score of 87
```

Comma Delimited ASCII Files

Data stored in a comma delimited ASCII file can be imported easily into virtually any off-the-shelf database product or spreadsheet. Many programs use this format to exchange data.

This format is not the same as a text file that is saved in ASCII form. Comma delimited format supplies a simple mechanism for separating records and fields and for distinguishing data types.

The record is the data found on a single line.

The field is the data that is found between commas (comma delimited) on the line. There is no fixed length for each field, and there is no fixed length for the records in the file.

Notice in Table 3 below, there are labels for each column. Each column corresponds to a field. Each row in the table corresponds to a record. In the conversion of this table to a comma-delimited file, the name of each field is gone. Only the field data itself is in the file. Position therefore becomes critical in determining the meaning of a field.

Notice also that empty fields, or blank fields may have to exist in the comma delimited file because the information is position dependent. In the third record there are two blank fields. The first is an empty number field, and the second is an empty text field.

Table 3: Example Table

Assignable Unit ID	Title	Type	Max Score	Duration	File Name
777APU-1	Auxiliary Power Unit	Tutorial	38	00:18:00	APU.EXE
777EL-1	Electrical Power, Part 1	Tutorial	41	00:23:00	ELEC1.EXE
777EL-2	Electrical Power, Part 2	Practice			ELEC2.EXE

Table 4: Comma Delimited File with Same Contents

```
"777APU-1","Auxiliary Power Unit","Tutorial",38,"00:18:00","APU.EXE"
"777EL-1","Electrical Power, Part 1","Tutorial",41,"00:23:00","ELEC1.EXE"
"777EL-2","Electrical Power, Part 2","Practice",,"","ELEC2.EXE"
```

CMI/CBT-LESSON COMMUNICATION

CMI and 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.

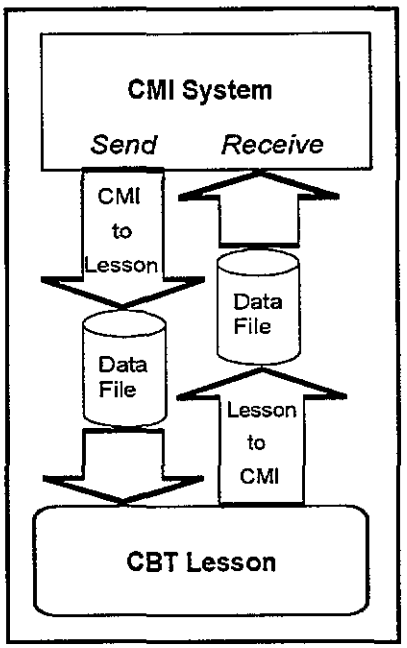

Figure 3: Communication Data Files

The 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.

CMI to CBT File

This is information that a typical lesson obtains from a CMI system to enable it to perform the functions expected of it. In Table 5, core items are listed first, followed by the optional group names alphabetically. (In the file, group names may be in any order.) After each group name are the keywords (if any) which are appropriate for that group. (In the file, keywords may appear in any order inside their group.)

A core item is one which must always be provided by the CMI system to be AICC compliant. Core items are those which a lesson may always depend upon being available. The lesson may or may not use the core items, but they are available if required.

Optional items are group and keyword data which may be needed by a lesson to perform optimally. However, the lesson must be constructed such that there is a default to be used if these optional items are not provided by the CMI system.

Table 5: CMI to CBT File Contents

Group Names and Keywords	Function of Group
[Core] Student_ID Lesson_location Student_Name Lesson_Status Output_File Score Lesson_Mode Time	Information required to be furnished by all CMI systems. What all lessons may depend upon at start up, from any AICC compliant CMI system.
[Core_Lesson] data is undefined and may be unique to each lesson	Information held by the CMI system for the lesson since last student attempt.
[Core_Vendor] data is undefined and may be unique to each vendor	Required information for some lessons. Must be furnished by CMI system.
[Comments] no key words <delimited>	E-Mail type information that an instructor or administrator wants to send to a student.
[Evaluation] Course_ID Comments_file Interactions_file Objectives_status_file Path_file Performance_file	File names and locations where the lesson should store the lesson evaluation information.
[Objectives_Status] J_ID.01 J_Score.01 Local_ID.01 J_Status.01	Information on each objective in an assignable unit.
[Private_Area] Path	Area where lesson can find and or store lesson-unique data.
[Student_Data] Attempt_Number Cumulative_Time Mastery_Score Max_Time_Allowed Time_Limit_Action Lesson_Status.01	Information on student performance expectations.
[Student_Demographics] Age Job_Title Birth_Date Language City Native_Language Class Race Company Religion Country Sex Experience State Familiar_ Name Street_Address Instructor_ Name Telephone Years_Experience	Personal information on student. Characteristics relating to student before course entry.
[Student_Preferences] Audio Text_Color Bookmarks Text_Location Lesson_Type Text_Size Text Video Window.01	Student selected options collected in previous lessons, or previous instances of this lesson.

CBT Lesson to CMI File

This is information that a lesson must/may make available to a CMI system. The core items (which the lesson MUST make available) are first, followed by the optional items listed alphabetically. Starting this file should be the first thing done by the lesson after launch

Table 6: Lesson to CMI File Contents

Group Names and Keywords	Function of Group
[Core] Lesson_Location Lesson_Status Score Time	Information required by the CMI system to function.
[Core_Lesson] data is undefined and may be unique to each lesson	Information required by the lesson. Passed to the CMI system to hold and return at the next start-up
[Comments] no key words <delimited>	Student comments on lesson.
[Objectives_Status] J_ID.01 Local_ID.01 J_Score.01 J_Status.01	Information on objectives contained in the lesson.
[Student_Preferences] Audio Bookmarks Lesson_Type Text Text_Color Text_Location Text_Size Video Window.01	Student selected options to be passed to next lesson he enters.

COURSE STRUCTURE DATA

The purpose of defining a CMI structure interchange format, is to simplify the process of moving a course from one system to another.

After moving a course, a review-and-modify effort is going to be required. The existence of standard interchange files however, should eliminate a large number of the manhours necessary to input a new course from scratch.

Basic Concepts

The files containing the structure of a course need to answer the question, "What information does a CMI system need, to present the training material to the student in the way desired by the designer?"

The approach taken by AICC guidelines assumes that the answer can be *implied* in a table that contains all of the lessons and lesson groups in a course.

The answer can be made *explicit* by stating prerequisites for each lesson (or assignable unit) in the course. When pre-conditions are set that must be met before a student can select or be assigned a lesson, each lesson, assumes a place in the course structure.

For instance, assume there is a course of six lessons. The order of the lessons can be implied by putting them in a simple table (Table 7); then reading the table left to right, and top to bottom.

To make this order explicit, assume lesson 6 has a prerequisite of the student having completed lesson 5, and lesson 5 requires passing lesson 4, and lesson 4 requires completion of lesson 3, etc. (Shown in Figure 4) This results in the linear presentation of the lessons in sequence from 1 through 6.

Table 7: Course Hierarchy Table

Root	Lesson 1	Lesson 2	Lesson 3	Lesson 4	Lesson 5	Lesson 6

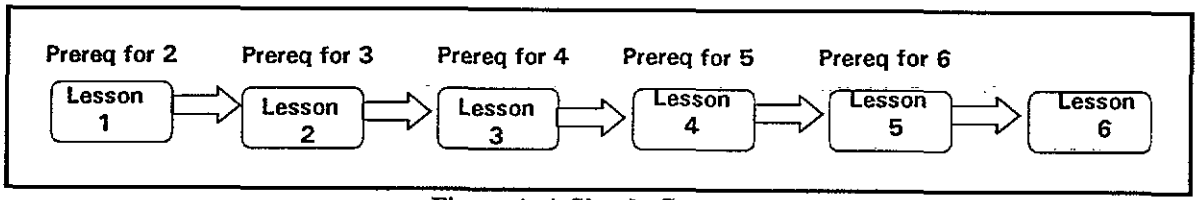

Figure 4: A Simple Course

In the AICC approach, prerequisites can be defined in terms of completed lessons, or *mastered objectives*. *Table 8 reflects* the prerequisites shown in Figure 4.

Table 8: Prerequisite Table

Assignable Unit	Prerequisites
Lesson 1	None
Lesson 2	Lesson 1
Lesson 3	Lesson 2
Lesson 4	Lesson 3
Lesson 5	Lesson 4
Lesson 6	Lesson 5

Of course, even with prerequisites there are cases where it is desirable to let the student chose the order in which he attempts some lessons. If three lessons have exactly the same prerequisites, then the student has an option -- after meeting the prerequisites -- of selecting any of the three.

In addition to files describing the course hierarchy and prerequisites there need to be files describing the elements in the course. This is textual information and not required to determine the order in which the student can take the course material. This information includes the titles of the various items in the course and a narrative description of them when desired.

Levels of Complexity

The AICC guidelines define five levels of complexity in describing the course structure. Increasing the level of complexity from level 1 to 2 to 3 to 4 to 5 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.

There are up to seven files that can be used to describe a course's content and structure. The level of complexity determines the number of files required and the amount of information required in each file. The following sections briefly describe the contents or purpose of each file. Tables 9 through 17 identify the names of each field, keyword or group in each of the seven possible files.

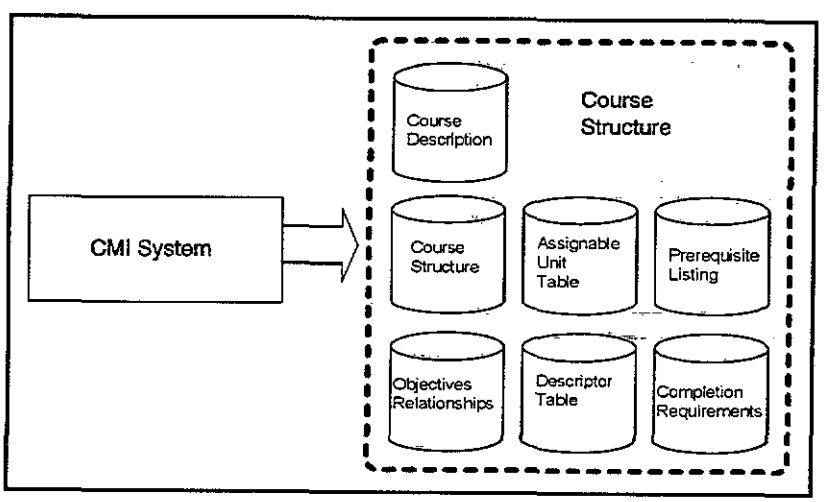

Figure 5: Course Structure Files

Course Description File (Table 9)

This file contains information about the course as a whole. It contains a textual description of the course, and general makeup of the course -- the number and type of elements.

Table 9: Course Description File

Groups and Keywords	
[Course]	
Course_ID	Total_AUs
Course_Title	Total_Blocks
Level	Total_Objectives
Course_Creator	Total_Complex_Obj
	Max_Fields_CST
Course_System	Max_Fields_ORT
[Course_Description]	

Assignable Unit Table (Table 10)

This file contains information about the assignable units (AUs) in the course. Each assignable unit 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 assignable unit.

Descriptor Table (Table 11)

This file contains a complete list of every course element in the course. Course elements include:

- Assignable Units
- 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. This file also contains any textual description created for an element in the course.

Course Structure Table (Table 12)

This file contains the basic data on the structure of the course. It includes all of the assignable units and blocks in the course, showing how they are organized -- which AUs are members of which blocks. And finally, it implies the order in which these should be taken.

Table 10: Assignable Unit Table -- the fields

System ID	Type	Command Line	Duration	File Name	Max Score	Mastery Score	System Vendor	Core Vendor

Table 11: Descriptor File -- the fields

System ID (for course element)	Developer ID (for course element)	Title	Line number	Description

Table 12: Course Structure Table

Block	Members -- Assignable units & other blocks			
Root	System ID	System ID	System ID	
System ID	System ID	System ID	System ID	System ID
System ID	System ID	System ID		

Table 13: Objectives Relationships Table

Structure Element	Members: Assignable units, blocks, & objectives			
System ID	System ID	System ID	System ID	System ID
System ID	System ID	System ID		
System ID	System ID	System ID	System ID	
System ID	System ID	System ID	System ID	System ID
System ID	System ID	System ID		

Objectives Relationships File (Table 13)

Objectives have complex and variable relationships to other elements of a course. For instance, a lesson may cover several objectives or a single objective may require mastery of several lessons. Other objectives may require the mastery of many sub-objectives.

The Objectives Relationships file is able to define all of these relationships. This file is optional, depending on the level of the course description.

Prerequisite Listing (Tables 14, 15, and 16)

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 assignable unit (AU) in a course.

There are three levels of complexity that may be used in describing prerequisites. The first (Table 14) allows a single prerequisite AU or block to be defined for each element in the course. The second (Table 15), allows prerequisites to be defined in the form of a logic statement (with "ands" and "ors") that includes objectives. The third (Table 16), and most complex prerequisite listing allows the definition of prerequisites for each mode in which the lesson may be used. Possible modes are:

- Review
- Browse
- Normal

Level 2

Table 14: Prerequisite Listing

Structure Element (Block or AU)	Prerequisite (Block or AU)
System ID	System ID
System ID	System ID
System ID	System ID

Level 3, 4

Table 15: Prerequisite Listing -- More Complex

Structure Element (Block or AU)	Prerequisite Logic Statement (Blk, AU or Obj)
System ID	System ID & System ID
System ID	System ID | System ID
System ID	System ID
System ID	System ID & (System ID | System ID)

Level 5

Table 16: Prerequisite Listing -- Most Comprehensive

Structure Element (Block or AU)	Prerequisite Logic Statement (Blk, AU or Obj)	Mode
System ID	System ID & System ID	N
System ID	System ID | System ID	B
System ID	System ID	R
System ID	System ID & (System ID | System ID)	N

Level 5

Table 17: Completion Requirements File

Block or Complex Objective	Completion Logic Statement (Blk, AU or Obj)
System ID	System ID & System ID
System ID	2*(System ID | System ID | System ID)
System ID	System ID
System ID	System ID & (System ID | System ID)

Completion Requirements (Table 17)

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.

Lesson evaluation data is contained in several files. The 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.

With lesson evaluation data, analysis tools and CMI systems are able to assemble information on multiple lessons, multiple uses of the same lesson, and multiple students.

The analysis of the information is not the subject of these guidelines. What is covered here is essentially raw data.

All of these files are optional. Up to five of them may be required to store all of the information desired from a CBT lesson.

LESSON EVALUATION DATA

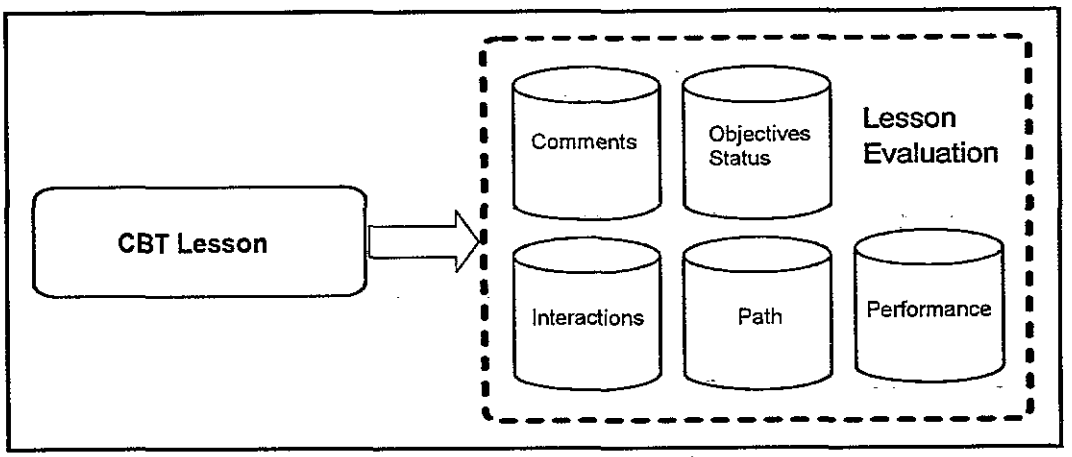

Figure 6: Lesson Evaluation Files

Table 18: Comments File -- the fields

Course ID	Lesson ID	Date	Time	Location	Line Number	Comment

Comments File (Table 18)

This is a journal file that contains freeform feedback from the student. It contains his criticisms and complements -- recorded as he moves through the lesson. It is a duplicate of the [Comments] group that is passed to the CMI system in the CBT-to-CMI file.

Interactions File (Table 19)

In this context, an interaction is a recognized and recordable input or group of inputs from the student to the computer. All of the items in this file are related to a recognized and recorded input from the student (or lesson user.)

Most commonly, these interaction records will be student responses to questions. The types of questions with defined response types are:

- True/False
- Multiple choice
- Fill in the blank
- Matching
- Simple performance
- Sequencing
- Likert
- Unique

Objectives Status (Table 20)

This file contains comprehensive information on objectives, including their ID and their status (passed, failed, or not attempted.)

Table 19: Interactions File -- the fields

Course ID	Lesson ID	Date	Time	Interaction ID	Objective ID	Type interaction	Correct response	Student response	Result	Weighting	Latency

Table 20: Objectives Status -- the fields

Course ID	Lesson ID	Date	Time	Objective ID	Local ID	Score	Status	Mastery time

Path File (Table 21)

This file allows an analysis of what path the student took through a lesson. It enables the analyst to determine when the student asked for help, when he selected alternative branches, if he selected optional instruction, and the order in which he proceeded through the lesson.

Table 21: Path File – the fields

Course IDs	Lesson ID	Date	Time	Element Location	Status	Why Left	Time in Element

ADDITIONAL INFORMATION

To obtain the complete documentation of the AICC standard, or to get more information on the AICC, contact:

Scott Bergstrom
AICC Administrator
University of North Dakota
Box 8216, University Station
Grand Forks, ND 58202-8216

Telephone: (701) 777-4380

GLOSSARY

argument Keyword argument. The information relating to a keyword that appears to the right of the equal sign. Also called keyword value or keyword data.

assignable unit The smallest element of instruction or testing to which a student may be routed by a CMI system. It is the smallest unit the CMI system assigns and tracks.

A program or lesson launched by the CMI system.

AU Abbrieviation for "assignable unit."

block An arbitrarily defined grouping of course components. Blocks are composed of related assignable units or other blocks.

This is a term used in the AICC document CMI Guidelines for Interoperability. A block may correspond to any level of the AICC instructional hierarchy above lesson, up to and including course.

bookmark Identification of a location in a lesson to which a student plans to return. Bookmarks are placed by the student for his own reference and review purposes.

CAI Computer-aided Instruction. Sometimes Computer-assisted Instruction. Normally used as a synonym for CBT. However some make the following distinction between CAI and CBT.

CAI (cont.) CAI: The computer as an aid to learning. Supports instruction, but is not the prime medium for delivery of instruction. Uses include presentation or practice but not both.

CBT: Computer as the primary mode of instruction.

CBT Computer-Based Training. The use of computers to provide an interactive instructional experience. Also referred to as CAI (Computer Assisted Instruction), CAL (Computer-aided Learning), CBE (Computer Based Education), CBI (Computer-based Instruction), etc.

CMI Computer-Managed Instruction has several definitions. In its broadest sense, it includes the following:

1) Rostering and storing student information.

2) Scheduling students and resources.

3) Computer acquisition and storage of student performance data. This is frequently referred to as student data collection instead of CMI.

4) Data presentation. After the data has been collected, it can be massaged by the computer, providing meaningful summaries for human interpretation. This is frequently referred to as data analysis instead of CMI.

CMI (cont.)

5) And finally, the computer can make decisions based on its analysis of the student's performance. It can manage the student's learning. It makes decisions as to what material the student should cover next, what material is not necessary, and what remedial actions if any, should be taken.

In some contexts, the term CMI excludes data collection and data analysis. The strictest definition of CMI includes only the fifth aspect, the computer management of the student.

The combination of items 3) and 4) above, is frequently referred to as "Student Evaluation."

core item

Data in a file for CMI/Lesson communication. A core item is one which must always be provided to be AICC compliant. Core items are those which a lesson may always depend upon being available. The lesson may or may not use the core items, but they are available if required. Most core items are in a single group entitled "core" (or "CORE" or "Core").

course

A complete unit of training. A course generally represents what a student needs to know in order to perform a set of related skills or master a related body of knowledge.

course (cont.)

Level 2 in the AICC Hierarchy of CBT Components:

1. Curriculum
2. Course
3. Chapter
4. Subchapter
5. Module
6. Lesson
7. Topic
8. Sequence
9. Frame
10. Object

course elements

Three items which constitute the building blocks for a course description. Each of these building blocks has its own title and attributes.

- Assignable Unit (lesson)
- Block, and
- Objective.

curriculum

A grouping of related courses.

Level 1 in the AICC Hierarchy of CBT Components:

1. Curriculum
2. Course
3. Chapter
4. Subchapter
5. Module
6. Lesson
7. Topic
8. Sequence
9. Frame
10. Object

demographics

Information associated with a student prior to entering a course. Student attributes. Typical demographic data includes the student's name, age, sex, years of experience, and native language.

group A unit of information in a standardized file for storing CMI information. Groups are large data items, generally several lines in length. A group extends from the group identifier to the next group identifier, and may include multiple lines. All carraige returns and symbols between group identifiers may be significant, depending on the definition of the specific group. Although groups may contain keywords, they may not contain other groups.

hierarchy The structure of lessons and/or courses which, to a large extent, determines how the student will perceive the course organization and in what order his lessons will be assigned.

interaction An exchange between a student and a program, beginning with a screen touch, a mouse click, a keyboard, or other input by a student, followed by an on-screen reaction of the program.

In the context of the CMI guideline for storing student performance data: A recognized and recordable input or group of inputs from the student to the computer.

item analysis. This can indicate how well an element of instruction trains; or how well a test question measures student performance. This enables quality control of the testing and instruction.

lesson A meaningful division of learning that is accomplished by a student in a continuous effort -- that is at one sitting. That part of the learning that is between designed breaks. Frequently requires approximately 20 minutes to an hour.
OR

A grouping of instruction that is controlled by a single executable computer program.
Or
A unit of training that is a logical division of a subchapter, chapter, or course.

Level 6 in the AICC Hierarchy of CBT Components:

1. Curriculum
2. Course
3. Chapter
4. Subchapter
5. Module
6. Lesson
7. Topic
8. Sequence
9. Frame
10. Object

lesson element An arbitrary division of an assignable unit that has been uniquely named (has its own ID). An assignable unit may have from two to hundreds of lesson elements.

keyword A unit of information in a standardized file for storing CMI information. Keywords are names of data items that are limited in size to a single line. This generally limits the data to 60 or 70 characters.

performance analysis Determination of a student's capabilities, based upon data collection of the student's interactions within one or more lessons. This helps to determine what the student knows, and what he learns. Comparing individual student progress with his peers gives a measurement of individual rate of learning.

router Software which sequences a series of lessons, tests, and other assignable units in a course. The router determines the order in which the student experiences segments of his computer-based training.

structure elements The parts of a course which can be uniquely assigned by a CMI system. These are units that can be rearranged to determine the order in which a student experiences a course of instruction. There are two structure elements in the AICC view of a course description:

- Assignable unit (the lesson)
- Block

value Keyword data. The information relating to a keyword that appears to the right of the equal sign. Also called keyword argument.