INTEGRATED LOGISTICS MANAGEMENT SYSTEM FOR THE BATTLE FORCE TACTICAL TRAINING PROGRAM

Dale Lotspeich
PHD, Naval Surface Warfare Center
Port Hueneme, California

Michael D. Kasmarik
Wunderlich - Malec Engineering, Inc.
Minnetonka, Minnesota

ABSTRACT

This paper presents an overview of a system design developed for the US Navy Battle Force Tactical Training (BFTT) program by a joint team of PHD NSWC, Eldyne, Inc. and RAC, Inc., of San Diego CA, Focus Learning Corporation of Pismo Beach, CA, and Wunderlich-Malec Engineering of Minnetonka, MN. This design provides a means to integrate all Interactive Courseware (ICW), Interactive Electronic Technical Manuals (IETM), and other required logistical support information into an electronic delivery system.

The system described utilizes Commercial Off-The-Shelf (COTS) hardware and software components to build the database and communications capabilities for the system, and includes the following capabilities:

1. An integrated methodology for defining and streamlining the development of logistical support information, principally IETM's and ICW.
2. Overall configuration management control and reliability / obsolescence of program components.
3. A version control method for authoring of IETM and ICW content, and for distribution of the IETM and ICW content runtime modules.
4. A repository system to make both IETM and ICW source level content objects and runtime content objects available to multiple locations.
5. A mechanism to connect to and retrieve information from various logistical support databases.
6. A network infrastructure to provide high performance network transport of all supported information to shipboard and land based sites, on demand.

The COTS technology is comprised of several applications. The foundation application provides a method to analyze and build a database of training requirements or objectives. Based on these requirements, training or performance support content is defined. This content is linked in the database to the requirements or objectives, and therefore provides a means to identify and maintain content when objectives or requirements change. In addition, linkages are provided to off-the-shelf ICW authoring packages, which in turn deliver actual IETM or ICW modules. Finally, a client/server messaging technique, successfully demonstrated in the commercial marketplace, is used to distribute and maintain version control of the IETM or ICW modules, and provides links to information from other logistical support databases. Current efforts for BFTT are focusing on the first four of the capabilities described, which are configuration management and definition, and development of IETM and ICW modules. Integration of the delivery technology will come at a later date.

Dale Lotspeich is an Acquisition Logistician with the Logistics Support Division of the Logistics Management Department, Port Hueneme Division, for the Naval Surface Warfare Center. Mr. Lotspeich graduated from Principia College, with a Bachelor of Arts in 1978, and from the Naval Material Command, Logistics Intern Program, in 1984. Mr. Lotspeich has worked on NDI/COTS programs since 1986, presently working on the Battle Force Tactical Training (BFTT- AN/USQ-T46(V)) program.

Michael D. Kasmarik is a Principle with Wunderlich-Malec Engineering, Inc., and holds a Bachelor of Science in Electrical Engineering from the University of Minnesota. Since 1975, Mr. Kasmarik has been involved in the design and management of research, engineering, and product development projects, including hardware, software, and network systems design, as well as manufacturing processes, computer based training, and on-line interactive performance support systems.

INTEGRATED LOGISTICS MANAGEMENT SYSTEM FOR THE BATTLE FORCE TACTICAL TRAINING PROGRAM

Dale Lotspeich
PHD, Naval Surface Warfare Center
Port Hueneme, California

Michael D. Kasmarik
Wunderlich - Malec Engineering, Inc.
Minnetonka, Minnesota

BACKGROUND

The mission of the Battle Force Tactical Training (BFTT) program is to provide combat team based simulation/training capability. This system is being applied over 140 ships in the fleet, as well as two shore sites and numerous home port sites. The hardware and software components that comprise the BFTT system will be continuously upgraded in such a manner that the design baseline is always expanding to keep up with technology.

As a result, the BFTT system is a dynamic process that is constantly changing. This then requires that the BFTT logistics support data be changed to stay in sync with the design and the released versions of the BFTT system. In addition, since the field upgrades and staged roll out of the BFTT system onto ships will be in groups, several valid versions of the system will be in place and operating at any time. This concept then requires a set of logistical support data, such as training, technical manuals, engineering drawings, supply support, provisioning, etc. to be maintained and updated on the same cycles as the BFTT system itself.

In order to cost effectively design, develop, review, revise and update all of the Interactive Logistics Support (ILS) data for a continually changing design, a new and fundamentally different approach to Logistics Support Data Management was required. This new system, called the Interactive Logistics Electronic Support System (**ILESS**), is the subject of this paper.

CONCEPT

When examining the various methods and technologies available to provide a dynamic data environment for ILESS, four major functional requirements were defined. It was determined the ILESS needed to be:

1. **Affordable over the implementation life cycle**
2. **Comprehensive, in terms of dealing with all type of logistical support data requirements**
3. **Accurate, by providing correct and up to date information**
4. **Real-time, maintaining and providing the most current data**

In examining these four requirements, it became clear that an interactive electronic media information system, rather than a paper or CD-ROM based approach was required. The challenge was to identify, within the range of technologies potentially applicable, any that would meet these requirements, and then define a means to integrate those technologies together to build the system.

From the functional requirements, it was determined that another level of specification defining the implementation requirements was needed. These implementation requirements focused on what the technologies must do to actually deliver the logistics data, and how this system could be deployed and maintained over its life cycle. The implementation requirements were therefore defined as follows:

1. All applications within the ILESS infrastructure needed to be capable of cost effective life cycle maintenance and support, with a minimum number of personnel required in support roles. This led to a requirement that the ILESS applications be tightly integrated, and user friendly.

2. Since the actual information needed for logistical support can come from a wide variety of existing systems, and in different formats, the ILESS needed to serve as a homogenous environment for storage, access, and distribution of all support information, as well as having the capability to provide connections to external systems.

3. The ILESS has to operate reliably over a variety of existing electronic delivery networks, both local area (LAN), and Wide Area (WAN).

4. The ILESS had to be capable of automatically distributing the correct “current” version of support information to the appropriate locations, on demand.

5. The protocol used to manage the ILESS information flows needed to be conventional, standard, and non-interfering with other information protocols potentially using the same network infrastructure.

6. The ILESS needed to be capable of scaling to a large number of access nodes, as the BFTT deployment reaches its intended target levels in the fleet.

ILESS ARCHITECTURE

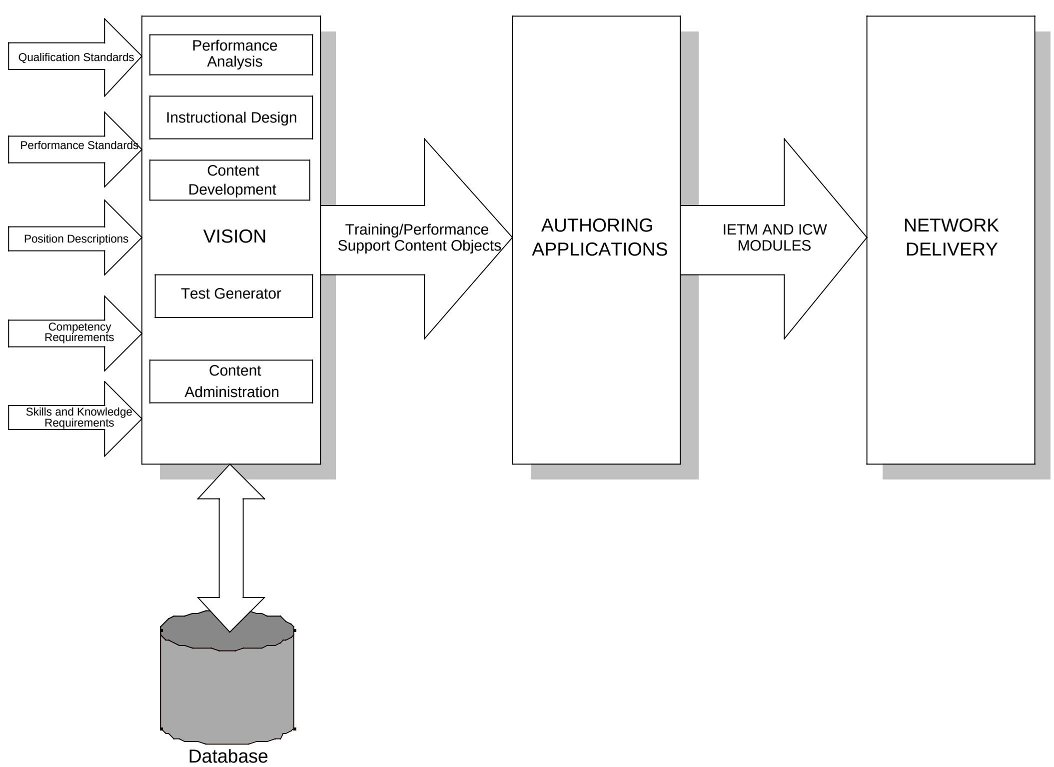
From the requirements described above, the ILESS model was developed, and consists of three COTS components (refer to Figure 1):

- **VISION, a product that provides tools for analysis of training requirements and objectives, as well as tools for the development of training and performance support content.**

- **A suite of authoring applications such as IconAuthor™ and AIMSS, for translating the content defined and developed in VISION, into interactive technical manuals (IETM’s) or interactive courseware modules (ICW).**

- **Client/server network technology to allow automated, version controlled distribution of the completed training or performance support modules, and connectivity to external data sources.**

Figure 1 - ILESS Model



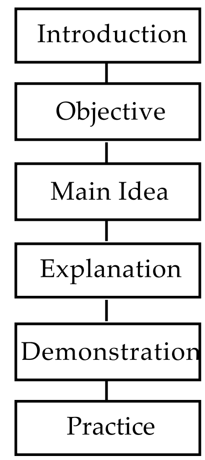
VISION

The foundation component of the ILESS model is the VISION product, which consists of five modules, each designed to streamline a portion of the content development process. Each module is discussed below.

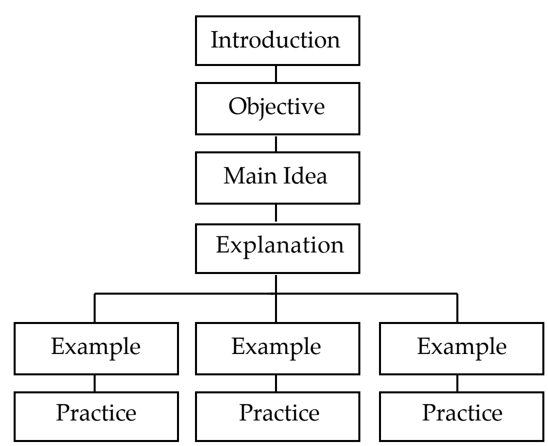
The Performance Analyst module assists users to analyze a job, work process, or system. Through the use of "templates" and "advisors", the user is prompted at each level of the analysis to identify the work structure, tasks, major steps, performance standards, and critical skill, knowledge, and ability components. The results are stored in a central database, and are presented to the user in a graphical hierarchy format that makes it easy to understand how the work components are related. From this analysis data, VISION can generate performance support documentation such as job qualification standards, hands on performance checklists, task analysis review forms, position descriptions, and work procedures, which through linkage mechanisms to authoring software can be used to develop IETM's and ICW.

The Instructional Designer module utilizes the analysis data to expedite the development of required instruction content. The user is presented with a graphical interface which displays previously entered job tasks, in list format. From this list the user selects a job task for which training is required. The user is then prompted to identify instructional objectives based on the performance standards and competencies for the selected task. Instructional Designer utilizes specific principles of learning models that are suggested according to the analysis and resulting classification of objectives. The available classifications of objectives and a block diagram of each are described below:

Procedure - A procedure objective applies to a job task that is a set of sequential steps that are always performed in the same way, and have the same outcome. The demonstration of the task is done in only one way.

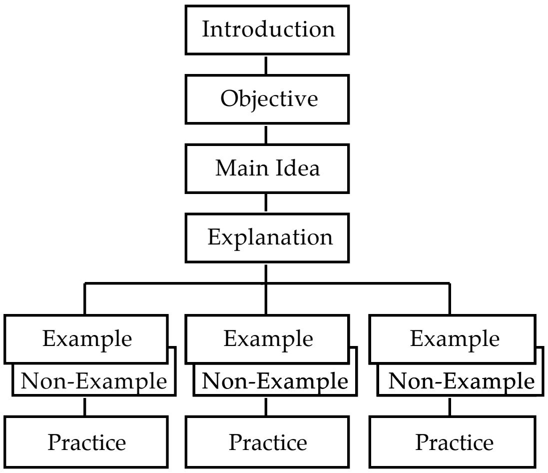


Rule - A rule objective works with a set of operations whose application and results may vary, providing different outcomes depending upon the situation. Successful demonstration of the objectives requires a variety of examples. The examples are usually presented in sequence from simple to more complex.



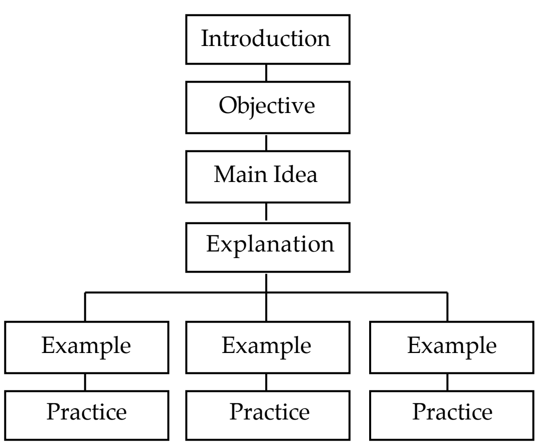
Practice may be presented in random order and could be broken down into specific critical attributes of the rule.

Concept - A concept is a group of elements where each element shares a unique set of characteristics common to the group. A concept is explained best by showing examples and non-examples.

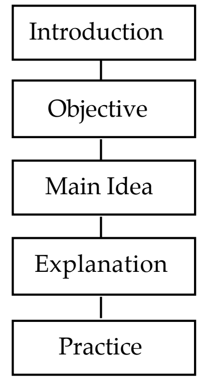


Practice would include the identification of examples and non-examples presented in random order. Details about what characteristic of the example makes it conform, or not conform, to the concept may be asked.

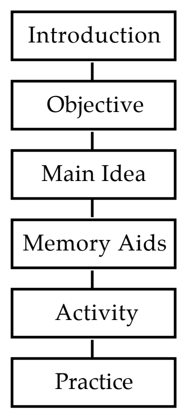
Process - A process objective is a set of sequential actions or events that occur within a system resulting in something being accomplished. To successfully apply a process task a person needs to be able to predict the likely changes to the system based on how the process works.



Principle - A principle is based on a cause and effect relationship that controls events and outcomes. A principle can be generalized to solve difficult problems for which specific procedures do not exist.



Structure - A structure consists of various parts, elements, or components. To successfully use a structure requires knowledge of the functions of the parts and how they fit together. The structure may be physical or abstract.



Fact - A fact is an arbitrary relationship among objects, symbols, or events that cannot be generalized or used to explain anything and can only be demonstrated by recalling the information.

Based on the user's selections, Instructional Designer generates an objectives hierarchy, which is represented on the screen as a "tree" structure. Once this hierarchy is prepared, context sensitive "advisors" provide rules, examples, and tips that help users restate the tasks and competencies as complete objectives. This information is then added to the "tree". The "advisors" also help to assign a classification to each objective that will later be used to suggest an appropriate instructional strategy.

Once the objectives are prepared, Instructional Designer helps the user to group and sequence them into instructional units, such as courses and lessons.

The Content Developer provides a rapid development of high quality stand alone lessons, instructor guides, and other performance support materials. Built-in lesson shells and content templates speed up the development process and promote better instructional design and greater consistency. Embedded instructional "advisors" guide users through every step of the development process to ensure that even novice authors or subject matter experts can produce high quality, efficient instruction or support material.

All shells, templates, and advisors are based on proven instructional design models. Through it's linkage to authoring applications, this instructional content forms the basis for IETM and ICW development, by providing a "storyboard" which is then enhanced with graphics, animations, and other media rich content to form a high quality interactive training module. The storyboards are actual shells containing text (in Rich Text Format), that can be either viewed on screen, printed as a design/review tool, or imported into various authoring programs where they become frame sequences. As frame sequences in authoring programs, they provide the basic structure for navigation and as a framework for building a complete content object.

The Test Generator assists users to develop test items and generate tests according to user defined parameters. The user is assisted in entering test items for each objective, resulting in an integrated test item database. Each test item in the database is linked to a corresponding objective.

The Test Generator then allows users to configure a test based on instructional parameters entered through a series of user friendly prompts and filters. These instructional parameters are based on the objective classification model defined in the Instructional Design process. The system draws items within the parameters randomly to produce the required test. The module can also be linked to an ICW module, which provides the explanations, examples, and non-examples, then presents the test.

The Content Administrator coordinates access and control of Vision developed content. A learner can search through the training database to select a subject area, then request the content to de delivered to the learners computer via a network connection for execution. This content can be an ICW module that is linked to the training objective in the VISION database, or an IETM. Version control is maintained, so that if a learner had accessed a module previously, and it was resident on his or her local computer, the next time that module is executed, it will check the VISION database to make sure it is still the current version. If it is current, it will execute. If not, the new version will automatically be transferred to the learners computer.

AUTHORING APPLICATIONS

As described above, once training objectives have been defined and exist in the VISION database, the content developed within VISION can be exported and/or linked to other media-rich content such as graphics, animations, simulations, digital video, and digital audio. By using interactive multimedia authoring tools, or IETM development tools, high quality interactive modules are built using the content exported from VISION as a structure or template for building the interactive modules.

An important feature of the ILESS model is the design of the templates for content development. In many instances, IETM's and ICW are developed as large monolithic executables, using CD-ROM as the distribution mechanism. These monolithic executables can be 10's to 100's of megabytes in size, and therefore do not lend themselves to efficient electronic network distribution. This is particularly true over a wide area network where bandwidth is typically an order of magnitude less than in a local area network. In addition, large monolithic multimedia executables are cost prohibitive to maintain, due to the time it takes to find, within their embedded structure, the frames or screens that need revision.

An ILESS, IETM or ICW module consists of multiple small, size constrained content objects, linked together to form a complete module. No content object is larger than 500 Kilobytes, and a variety of compression technologies are applied to the media-rich content components such as graphics and animation, to maximize the amount of information included within each content object. A "main" module controls the order of execution and organization of the individual content objects, so that to the user the application looks and acts monolithic. This technique allows a tight integration of IETM and ICW, since each consists of content objects, all content is available from within either type of application.

This model also allows the content objects to be distributed individually and efficiently via the ILESS network infrastructure, utilizing a minimum amount of network bandwidth. Version control is also simplified, since the "main" program checks version on all of it's content objects prior to execution, and replaces only those objects that are out of date.

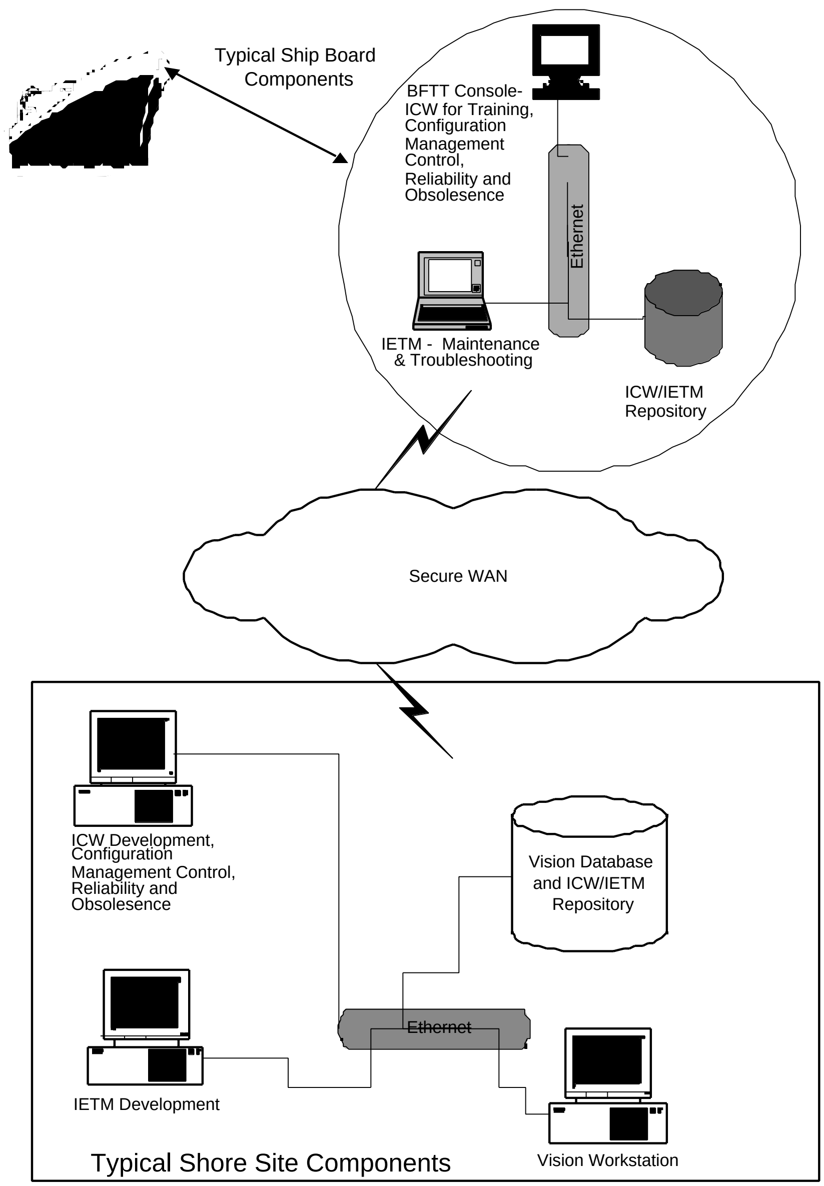
Finally, since these content objects are linked to objectives in the VISION database, through VISION it requires a minimum of effort to find and revise or update content objects when training requirements change.

NETWORK DELIVERY

The ILESS network delivery mechanism is characterized as a client driven architecture. Rather than having servers maintain complex databases of who has what ILESS material, or deal with the scheduling of updates to large numbers of remote clients, the ILESS applications include capabilities that upon start of execution check version with host servers, and request updates only when required.

Finally, the ILESS model allows connections to other sources of data. While this capability has been demonstrated, the current BFTT implementation plans are focusing on ICW and IETM development. External connectivity will be addressed in future phases of ILESS implementation. Shown in Figure 2 are typical physical components of ILESS. The shipboard VISION Database and ICW/IETM Repository provide a means to localize ILESS ICW and IETM content for shipboard use. Version control on this machine can be synchronized with a shore based content server periodically to ensure that the shipboard information is current.

Figure 2 - Typical ILESS Components



SUMMARY

All of the ILESS capabilities described are provided by a selection and integration of Commercial-Off-The-Shelf (COTS) products and technologies.

Major features of the ILESS Model are utilization of an instructional design application to facilitate integration of IETM and ICW content, the ability to cost effectively provide life cycle maintenance of ICW and IETM content, and the ability to electronically distribute ICW and IETM content to remote locations.

It is expected that ILESS can provide logistical support information at a fraction of the cost of custom, or conventional client/server products or architectures.