TAKING ADVANTAGE OF LOW-COST COTS SOFTWARE FOR THE DEVELOPMENT OF TRAINING MANAGEMENT SHELLS

Ellen E. Shay and Linda L. Terlecki
Loral WDL
Hanover, Maryland

ABSTRACT

In the past few years, a wide variety of powerful, low-cost commercial off-the-shelf (COTS) software packages have been released, allowing users to build complex applications using minimal programming skills. These packages have made graphical user interfaces particularly simple to develop by providing robust on-line tools and support features. This allows applications to be quickly and easily prototyped for early user involvement, better user understanding, and overall proof of concept. Developers can concentrate on the requirements and design of an application, spending time on the "look and feel" of the application instead of the "how" because the how has been simplified.

Our requirements were to build a training system management shell that provided student logon, access to course materials, management of student data, and course evaluation data reporting. This shell was part of an overall effort to produce a general-use, Multimedia Personal Computer (MPC)-compliant platform that was also to be used for language enrichment materials. This platform included a specified set of hardware and COTS software. We analyzed the given set of software tools, then developed a strategy to enhance the overall training product by providing a training system management shell for a minimal investment. It was determined that the best strategy would be the use of the built-in capabilities of the provided COTS software. The training system management shell was developed with a minimal use of traditional software development procedures, focusing only on the essentials for successful user management in the specific environment.

We found this approach to be appropriate when it is necessary to enhance existing student management and course evaluation capabilities, integrate courses from different sources, minimize time/resources spent on non-instructional aspects of a project, accommodate a short development schedule, and/or utilize resources whose skill level and/or availability won't allow a traditional approch to development.

ABOUT THE AUTHORS

Ellen Shay is currently a Training System Engineer at Loral WDL. She has integrated hardware, software, and courseware for several training systems. Prior to joining Loral in 1984, she was employed by McDonnell Aircraft Company as a Logistic Element Manager for Training and an Instructional Designer. Ms. Shay holds an MS in Media Education from Florida State University and a BA in Computer Science from the University of Florida.

Linda Terlecki is currently the Project Engineer for the Job Skills Education Program, a basic occupational learning program at Loral WDL. She was the lead system engineer for the development of several training delivery systems deployed world-wide. Ms. Terlecki has been involved with the design and development of computer-based training systems since 1980. Prior to joining Loral, Ms. Terlecki was a software engineer with the Hazeltine Corporation. Ms. Terlecki holds a BA in Political Science from Dickinson College in Carlisle, PA.

TAKING ADVANTAGE OF LOW-COST COTS SOFTWARE FOR THE DEVELOPMENT OF TRAINING MANAGEMENT SHELLS

Ellen E. Shay and Linda L. Terlecki
Loral WDL
Hanover, Maryland

INTRODUCTION

Until recently, the development of computer-managed instruction (CMI) systems required that almost all components be built from scratch, including the database, user interface, and data reporting. This effort required a lot of specialized resources. In addition, the user interface wasn't particularly intuitive, given that development tools were unavailable and the project personnel were busy developing the database. Fortunately, the software industry has produced inexpensive, powerful applications that have revolutionized software development. As developers and users, we can now go to the local computer store and purchase much of the functionality our projects require. We may then spend our time customizing the product, making it easier for our customers to use.

This paper examines the current capabilities of these commercial off-the-shelf software (COTS) packages and how they were used to develop a training management shell.

USING COTS PACKAGES

Capabilities of COTS Packages

Require Minimal Programming - Recent COTS releases allow users to build complex applications using minimal programming skills. Spreadsheets are being turned into custom financial analysis programs and databases are becoming sophisticated inventory and tracking systems. COTS packages include tools that allow users to develop elaborate custom applications with minimal help from programmers. Tasks that used to require a lot of programming may now be provided as an option on a menu or button bar.

On-Line Tools and Support Features - Robust on-line tools and support features have made the user interface particularly simple to develop and customize. Frequently-used functions are easy to automate. These features assist and simplify the translation of paper-and-pencil designs into polished applications with all the standard Windows features, including buttons, pull-down lists, and graphic logos. Some packages have "Wizards" or "Writers" for screen forms and reports, making it easy to quickly generate a form or report from a query. These features ask a few questions, present options that represent the most common designs, then automatically generate the form or report. Graphical tools are used to draw forms and reports instead of programming. Macro languages allow developers to completely customize the look and operation of the underlying application—even to the point of replacing its menus, screens, and dialogue boxes. The end result is easier-to-use, custom-fit interfaces for users that are indistinguishable from a program written from scratch in a language such as C. Now that COTS packages are moving towards object-orientation, developers and users can select an object in a table, form, or report and set the options associated with that object. This allows the developers and users to make complicated changes to forms or reports very easily.

On-Line Helps - COTS packages now provide extensive on-line helps. Most packages offer context-sensitive help, giving you information specific to the task you are performing at the time you requested the help. Windows-based applications have included the capability to annotate Help files, customizing them. There are also packages that allow developers to build Help files for applications they build using COTS packages.

Issues in Using COTS Packages

There are some issues that should be considered when using COTS packages. There has been an explosion of software, especially software employing graphical user interfaces. The capabilities of the available software are changing almost as fast as hardware. If the press releases are to be believed, better and/or revised products are always just about to be released. Developers and users find that they must answer the question: Should I use the current version or wait some number of months for the next version or a new package? This question can cause schedule problems if the wrong decision is made. With the rapid increase in the number of COTS packages, developers and users must ensure they are aware of the pitfalls that may occur with beta releases and first versions of new software. If there are problems with the COTS package, your project will be at the mercy of the software company for fixes. You have no control over the timeliness of the fixes. On the other hand, it is their problem to fix and not yours.

Benefits of Using COTS Packages

The benefits of using COTS packages far outweigh any issues concerning COTS packages.

Allows Short Development Schedule - Using COTS packages accommodates a short development schedule. Most of the desired functionality is built-in or easily produced. COTS packages, for the most part, have already been through extensive testing—developers can concentrate on testing the modifications they made.

Concentration on "Look and Feel" vs. "How" - Developers can concentrate on the aspects of their project that would be difficult and/or time-consuming no matter how they were implemented. Developers are able to spend time on the requirements and design of the project because the built-in features of the COTS package shortens the development phase. They can spend their effort on the "look and feel" of their product instead of the "how" because the how has been simplified. The custom application can take advantage of all the high-level facilities provided by the underlying application. For example, an elaborate field-transfer protocol or calculation that might require hundreds of lines of code in a conventional language can be invoked with a single macro language function call.

User Interface is Easier to Develop - The user interface is easier to develop than in the past. Most COTS packages provide interface design tools and macro functions that specifically support user interface development. In particular, Windows applications take advantage of the graphical environment to allow end users and developers to create sophisticated applications with very little programming.

Using the power of the COTS features, developers can prototype applications such as a training management shell. Prototyping allows for early customer/user involvement with better understanding by the user. Developers may also try out different solutions to the requirements, proving their concepts. Once the product is delivered, the customer and/or user can customize and tailor the product, making the product maintainable by the customer.

Benefits End Users - COTS packages also benefit the end users by requiring less training and increasing productivity. First, many users are already familiar with many of the COTS packages on which the customized applications are based. Second, developers use features such as macro languages to hide the rows and columns of spreadsheets and to design document templates for word processing.

There is a wide variety of powerful, low-cost packages that are just as effective as many of the more expensive, complicated packages.

APPLICATION OF COTS TO A TRAINING MANAGEMENT SHELL

Overview of Project

Our main objective to produce a general-use, Multimedia Personal Computer (MPC)-compliant delivery platform, including hardware and software, that was also to be used to deliver language enrichment materials. We analyzed the given toolset, then developed a strategy to enhance the overall product by providing a training system management shell and data reporting for a minimal investment. However, the training system management shell was not to support an elaborate set of features or user interface.

In addition, we were tasked with converting existing courses that used an outdated interface and were running on outdated technology. The content and design of the courses were to stay the same. The hardware and software platform would change.

Requirements - During the development of the training management shell, the focus was on providing only the essentials for successful user management in the specific environment for the courses. The requirements included:

- User log on
- User progress data collection and backup
- Course evaluation (data analysis)
- Course access
- Access to full capabilities of the COTS packages delivered with the system (e.g., Microsoft Word)

Hardware and Software - The customer specified the hardware and software that was to be used as the delivery platform. The hardware suite, an MPC-compliant PC with interactive video, is defined in Table 1. A comprehensive list of COTS software is listed in Table 2.

Table 1 Hardware Suite

COMPONENT	DESCRIPTION
CPU	486-66MHz CPU with Dual Floppy Drives, 256K Cache, 32MB RAM
Hard Drive	660MB Hard Drive
Input Devices	Keyboard, Mouse
Monitor	19" Multisync Monitor
Media Sources	CD-ROM Drive, Videodisc Player
Graphics Cards	Graphics Adapter, Video Overlay
Audio	Digital Audio Card, Audio Mixer, Audio Amplifier, Audio Speakers

Table 2 Project Software

TYPE	PACKAGES
Operating System	DOS 5.0, Windows 3.1
Utilities	Virex 3.1, Norton Desktop Windows 2.0, Grafplus Screen Print
Authoring	Authorware Professional for Windows 1.1
Font Utilities	All Type, Adobe Type Manager, Kaballah Font, Fontographer
Development	Microsoft SDK C/C++ 7.0, Microsoft Access 1.0
Other	Microsoft Word for Windows 2.0

Rationale for the Approach

There were three factors that affected the approach we used in the development of the training management shell:

- The customer provided a specific suite of hardware and software
- The development schedule was short
- Minimal resources, including staffing, were available for the project

We considered several approaches before deciding to use the built-in capabilities of the provided COTS packages, with minimal coding.

Description of the Process

Overview - The functional requirements were specified during the definition of the user interface. The process was iterative, with overlap between the Analysis and Design phases and the Design and Development phases. The data collection requirements were derived from the reports specified. See Figure 1 for a graphic of the basic process.

Analysis - The development team, working with the customer, used flowcharts to define the concept of operations (CONOP) for the utilization of the system and the desired data analysis reports. (See Figure 2). The following is a brief description of the CONOP:

1. Student selects the icon of the desired language.
2. If the student is already registered, the student enters the course. If not, the student registers.

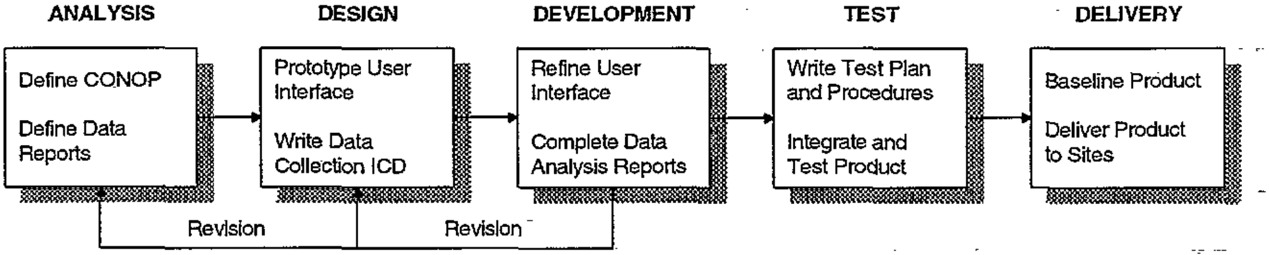
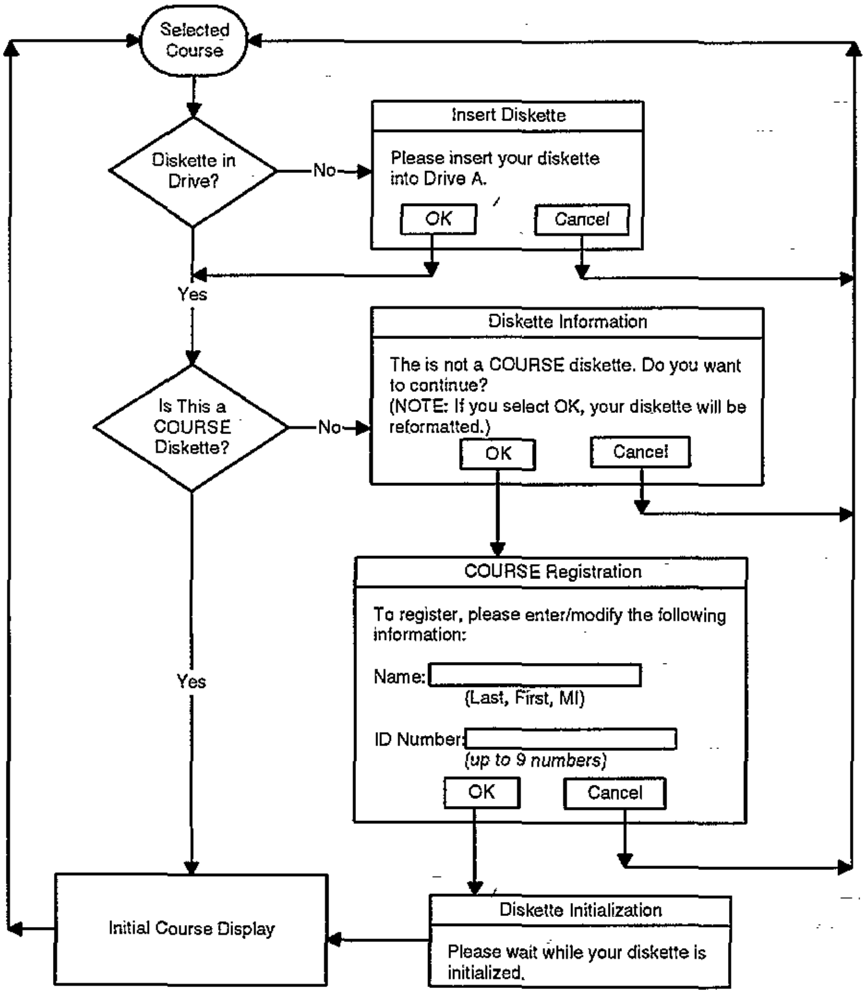


Figure 1 Training System Management Shell Development Process



Please insert your diskette into Drive A.

The is not a COURSE diskette. Do you want to continue?
(NOTE: If you select OK, your diskette will be reformatted.)

To register, please enter/modify the following information:

Please wait while your diskette is initialized.

Figure 2 CONOP Process Example

3. Progress data and test results are saved to the student's diskette.
4. When the course is completed, the student sends the diskette to the schoolhouse.
5. The schoolhouse imports the data.
6. The schoolhouse accesses data analysis reports.

Design - The functionality of the user interface and reports was designed using line drawings of the proposed displays. We produced a Project Implementation Plan that included the user interface, reports, and CONOP. The COTS software was then used to prototype and polish the user interface. Design reviews with the customer consisted of presentations of both the line drawings and on-line prototypes. The customer provided comments on the design, which was then revised for further review. This process resulted in a detailed design for implementation. The courseware and software Interface Control Document (ICD) was written for data structures and data collection. Due to the rapid prototyping, much of the interface for user management was complete by the end of the design phase.

Development - As each part of the shell was prototyped, it was demonstrated to the customer for feedback. The user management interface was refined and the course evaluation (data analysis) reports were completed.

Integration and Test - The Project Implementation Plan was used as the basis for test plans and procedures, making them easy to write and trace back to the requirements. Full functionality, data collection and the user interface were all tested. The COTS software operation did not need to be verified—we only had to ensure the training system management shell was fully integrated with the courseware and worked properly. The applications developed with COTS software were integrated and tested incrementally. The courseware developers and the customer were responsible for testing the converted course. Final integration and test verified that the training system management shell and converted courseware operated smoothly and could transfer data. The time spent during the analysis and design phases and the use of COTS products led to a high return. Integration and test was completed ahead of schedule, with only one anomaly.

Product Delivery - The courseware and software were baselined, loaded on the hardware, then delivered to the sites. The delivery consisted of an integrated system including the training system management shell, courseware, hardware, user manual, and student guide.

Examples of Displays

This section provides some examples of the user interface displays that were developed using COTS packages.

The screen shown in Figure 3 is the Language Enrichment Group. The student selects the course from this display.

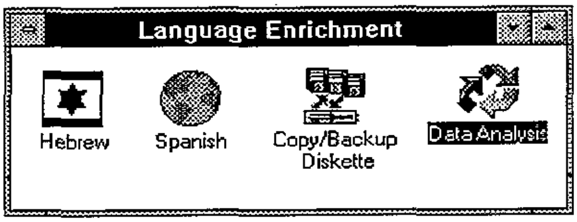
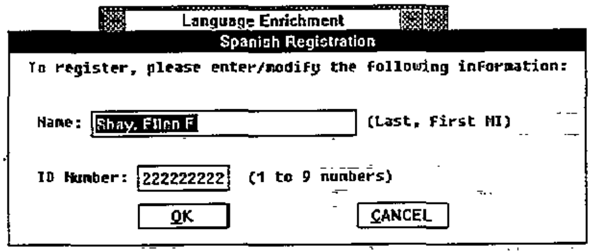


Figure 3 Language Enrichment Group

The screen shown in Figure 4 is used by the student to register for a course.



To register, please enter/modify the following information:

Name: Shay, Ellen F (Last, First MI)

ID Number: 222222222 (1 to 9 numbers)

Figure 4 Student Registration

The screen shown in Figure 5 is the Data Analysis Group. The schoolhouse selects the desired data analysis action.

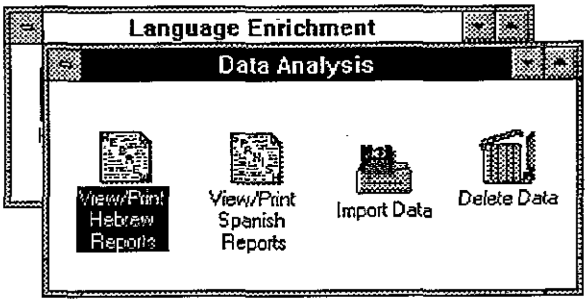
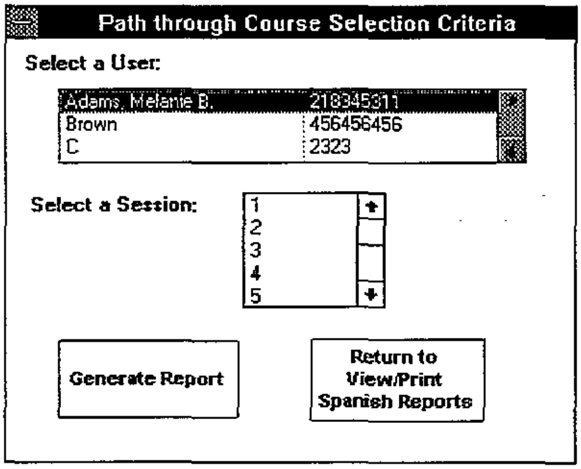


Figure 5 Data Analysis Group

The screen shown in Figure 6 allows the schoolhouse to specify the student(s) to report on.



Path through Course Selection Criteria

Select a User:

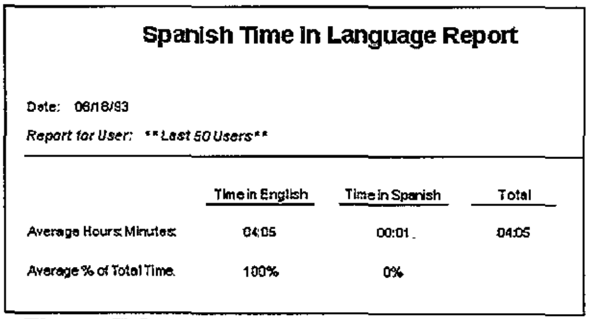
Adams, Melanie B.	218345311
Brown	456456456
C	2323

Select a Session:

1
2
3
4
5

Figure 6 Selection Criteria Screen

The screen shown in Figure 7 is an example of a report.



Spanish Time In Language Report

Date: 06/16/93

*Report for User: **Last 50 Users***

	Time in English	Time in Spanish	Total
Average Hours:Minutes:	04:05	00:01	04:05
Average % of Total Time:	100%	0%	

Figure 7 Sample Time in Language Report

SUMMARY

Results of Process

The use of COTS software for the development of the training system management shell produced two major results:

1. The requirements as defined by the customer were met to their satisfaction.
2. The development of the shell was completed four weeks earlier than originally estimated and required less re-work than more traditional development methods.

When to Use This Approach

Using COTS packages for training management shells is appropriate when a project requires:

- Enhancements to student management and/or course evaluation capabilities
- Integration of courses from different sources
- A minimum of time/resources spent on non-instructional aspects

On a more generic level, COTS packages should be considered when any project requires:

- A short development schedule
- That the existing resources be utilized, but the skill level and/or number of available resources will not allow a traditional approach

Benefits

Using COTS packages has several benefits. The product is easy to enhance and maintain and is self-documenting. Developers can concentrate on the requirements and design instead of the mechanics of implementation. Prototyping can be used to ensure customer understanding and to prove concepts. The basic COTS package features may be used to meet most users' needs. End users require less training and are more productive.

Some Final Thoughts

When developing training management shells, do not rule out COTS packages. They can be just as effective and much less expensive than more complicated approaches. Finally, Christine Comaford of PC Week put it well:

> "I used to program for a living. However, given a choice between solving a problem by writing some code or buying the solution off the shelf, I'd take the store route every time. So should you. With the variety of terrific tools available today, you can spend hundreds of dollars to save thousands more in unnecessary coding. Why not save your programming efforts for the critical pieces of your application development environment and buy the rest?"

REFERENCE

Comaford, C. (1993, February 15). Tools that simplify the GUI environment. PC Week, p. 63.