A WEB ARCHITECTURE TO SUPPORT THE TAILORING OF TRAINING CONTENT AND LEARNER ASSESSMENT

Roger Counihan
Justin M. DeVoge
Ellen J. Bass

Department of Systems and Information Engineering
University of Virginia
Charlottesville, VA 22903

ABSTRACT

Instructional systems designers can face a difficult task with training development including curriculum content, delivery method, and learner assessment. This situation becomes more complex when multiple types of learners must be trained with overlapping but not completely similar content. This paper discusses a flexible web-based architecture that supports the tailoring of training content and learner assessment. The architecture supports the creation of multiple training treatments and assessments, and learner performance data collection.

1 INTRODUCTION

Organizations often need training tailored for different types of employees. For example, new hires may need more in-depth training while experienced employees may only require updates on new or enhanced products and processes. In addition, even for a particular learner group, training content and assessment may require enhancements over time.

In some domains, there exist regulatory training requirements that mandate different training content based on job classification. In aviation, the Federal Aviation Administration (FAA) mandates different training for pilots, flight dispatchers, flight attendants, aircraft maintenance personnel, etc. Even within one group such as flight dispatchers, the FAA requires different content for initial certification, new hire, and recurrent training (FAA, n.d.). For example, Federal Aviation Regulation (FAR) Part 65 Subpart C defines the knowledge requirements for dispatcher certification training while Subpart N defines the requirements for new hire dispatcher training programs at an airline (FAA, n.d.) (Table 1). To meet different training requirements such as these, instructional systems designers would benefit from tools that support the tailoring of training content and learner assessment to classes of learners.

The popularity of web-based systems for the delivery of training has been steadily increasing as they are flexible and accessible to learners. Also, they can be quickly updated without having to physically send different organizations new materials. We have collaborated with major and regional airlines who agree that a flexible and re-usable web-based system for the delivery of training would be beneficial to their organizations, especially if both meteorological information and airline-specific training is included. For example, flight planning tools, weather products, fleet information, and other airline-specific considerations, such as routing and procedures are all topics that require different training between airlines. The airlines have also expressed the benefit of having a system that can be easily accessed, employed, and modified across organizations.

This paper discusses the operational requirements and architecture of a flexible web-based system for the delivery of training content and learner assessment. This paper also discusses the creation of a prototype web-based training system that implements this architecture using flight dispatcher icing-related training as the curriculum.

<table>
<thead>
<tr>
<th>Regulation</th>
<th>Training Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAR Part 65 Subpart C (Certification)</td>
<td>• Meteorology, interpretation of weather data, aeronautical decision making</td>
</tr>
</tbody>
</table>
| FAR Part 121 Subpart N (New Hire)    | • Meteorology, interpretation of weather data, aeronautical decision making  
|                     | • Aircraft specific training, airline emergency procedures |

2 OPERATIONAL ARCHITECTURE AND REQUIREMENTS

2.1 Training Content Related Requirements

An instructional system designer (ISD) may want to tailor training for groups of employees for different organizations based on their needs or constraints. He may also want to investigate the effectiveness of different training content or evaluation materials. This type of customization can be viewed as the creation of multiple “treatments”. Each “treatment” is comprised of a subset of the entire training curriculum.
The training curriculum within each treatment is organized as a set of training content and learner assessment components (modules). Each component is a set of pages. Training components are pages adapted from the ISD’s training material. Assessment components are comprised of knowledge evaluation questions covering the training materials. If the ISD is investigating training across learners, the learner assessment components may include a demographic questionnaire. Additionally, if the ISD wishes to evaluate the human-computer interface of the training material, there may be an assessment component consisting of usability questions.

Based on this operational concept, the training system must support the selection and organization of individualized training material. The system should support the mapping of content and assessments to a training treatment, thereby supporting the creation of a curriculum from a larger set of materials. Each resulting treatment can then be assigned to individual learners.

To make the system more modular, the system must support the mapping of individual training pages to training components, and questions to assessment components. This will give the ISD more flexibility to assign different sets of material to groups of learners.

To facilitate this process, the system must have access to the set of web-pages or files that make up the ISD’s training curriculum. This is necessary to deliver the materials to the learners.

The system should be able to display the training materials, knowledge assessments, and record the learner response associated with them. It should also store the learner responses to knowledge assessments, demographic forms, or usability questionnaires.

2.2 Training Session Related Requirements

When a learner logs into a training session, he views the training and assessment materials for his assigned treatment, in the order designated by ISD.

Once a learner has begun a treatment component, he views the pages associated with that component. As specified by the ISD, the pages can be presented in a sequence or in a random order. The ISD can also specify the recording of page and component viewing times to help recognize content that confuses learners or a page that may be difficult to understand. The learner can study the training material at his own pace and, after completing a component, can view his progress through the training, and the components left to be completed. A learner may complete the treatment in one or more sessions.

For assessment components, the questions may also be presented in a random or specified order, and be timed. As the learner completes the assessment components, the system stores the learner responses. The ISD can later access the recorded results of the knowledge assessments.

Based on this operational concept, the system must control the delivery of training to a learner and track his progress. The system should allow the learner to log into the training, in order to track his progress and verify his identity. Logging into the system is a common start to a training session that allows the system to know the identity of the learner.

The system should display components associated with a learner’s treatment, supporting the need to have learners view only the content that has been assigned to them. The system should display the correct pages for each component, to support the ISD’s need to organize content as components, enhancing flexibility and modularity. When developing components, the ISD will specify the order in which the components and individual web-pages should be delivered. The system must display the components and pages in this order. For treatment components, the ISD will typically have a specified order in which to deliver the training pages. For knowledge assessments, the ISD may also specify that questions occur in a random order, useful if a pre-test post-test assessment strategy is employed.

The system should time the duration of individual page and component views, allowing the ISD to identify pages which are complicated or content that confuses the learner. The system must track learner progress to facilitate keeping the ISD aware of the learner’s advancement through the training. The system shall also allow multiple training sessions for a learner, allowing a user to complete the training across multiple periods.

3 SYSTEM ARCHITECTURE

When compared to more tightly integrated architectures, such as client-server or fat-client architectures, a three-tier architecture provides increased performance, flexibility, maintainability, reusability, and scalability (Eckerson, 1995). This architecture follows a three-tier (data, application, user tier) design (Figure 1):

1. Data Management Tier
2. Application Logic Tier
3. Learner Interface Tier

3.1 Data Management Tier

The data management tier consists of a relational database that holds the structure of the training system, learner information, and assessment results. The data are stored in individual tables that represent the treatments, components, training pages, and learner specific data of the training system. The structure of the database is shown in Figure 2.

The ‘Treatment’ table stores the name and description of a treatment. Treatments are composed of one or more components, which are modules of training content or learner assessments.
Components are made up of a set of pages. In training content components, these pages are ordered by the ISD using the position field in the page table. For assessment components, the ISD can request a random presentation order by setting the RandomOrder boolean in the component table.

Treatments are connected to components via the ‘TreatmentComponent’ relation, which enables a component to be a part of multiple treatments. The relation also controls the presentation order of the components within a treatment.

The ‘Learner’ table holds the login and treatment information for a learner. A learner is assigned to one treatment, so the learner table is linked to the treatment table with the TreatmentID field. Two relations are required to track the learner’s progress. LearnerComponentProgress tracks and times the completion of components. LearnerPageProgress tracks and times the viewing of individual web-pages. The ISD designates that a question page requires learner input by setting the ReturnsData boolean in the page table. The learner’s responses to the pages are stored in the LearnerResults table.

3.2 Application Logic Tier

The application logic tier controls the delivery of training, tracks learner progress, and manages learner results. It is composed of server side scripts and session variables.

Every time a learner logs into a training session, a new “web-session” is started on the web-server. A web-session is a portion of memory dedicated to the interaction between the web-server and a learner. The web-server can monitor multiple sessions at the same time, allowing multiple learners to use the system concurrently.

3.2.1 Session Variables

The web-session stores session variables, which maintain the state of the training session. A session variable is a variable that can be accessed by programs on the web-server, to determine the current state of the system. The basic session variables are:

- Learner’s ID
- Treatment ID
- Current component
- Current page
- A sequence of random integers (for the presentation of content in a random order)
- Component start time
- Page start time

Session variables hold the required lookup information for the server to retrieve treatment and learner data from the data store using programs called “server-side scripts”.

Figure 1: System Architecture
Figure 2: Database Schema
3.2.2 Server-side scripts

A server-side script is a small program which is run on the web-server. It can dynamically generate web-pages, determine the correct web-page to be displayed, and writes information to the data store. The server side scripts contain the logic to monitor the learner’s progress and deliver the training material to the ISD’s specifications.

A login script allows a learner to log into the system. It verifies the learner login information against the database, and then creates and initializes the session variables. Holding the learner’s login allows the system to track the learner’s progress and store his results.

An index script creates a dynamic web-page that displays the learner’s progress. It shows the learner the training components completed and the components left to study. From the web-page created by the index script, learners can only access the next component in their training. This ensures that the learner views the components in the order specified by the ISD.

The display of materials is controlled by a set of scripts that are called at the start and finish of components and web-pages:

- **StartComponent** – This script updates the current component session variable to reflect that a learner has begun viewing a specific component. It also sets the page start time session variable to time the learner viewing the component.
- **EndComponent** – This script ends the timing for the component, and writes the time as well as the date completed to the database. After updating the database, the program returns to the Index script to display which component should be completed next.
- **StartPage** – This script sets the current page session variable to reflect that the learner has begun viewing a page. It also sets a session variable to time the learner viewing the page. Then it calls the correct web-page to view.
- **NextPage** – This script reads the current session variables, and queries the database to determine which page should be displayed next. The script writes this information to the session variables, and then displays the appropriate page.
- **EndPage** – This script ends the timing of the page, and also writes any learner input as a result of the page (such as the answer to a question) to the database.

3.2.3 File Management

In addition to running scripts, the web-server manages the pages holding training content. The web-server tracks the location of these pages and displays them to the learner when requested by a script. A folder structure organizes the training pages, and other files of the system (Figure 3). There are three main folders in the system:

- **Data** – This folder holds the training and assessment content.
- **Code** – This folder holds the code for the server-side scripts.
- **Docs** – This folder contains the documentation of the system and training curriculum. The information stored here is not used by the programs; the folder is solely for ISD convenience.

![Figure 3: Web-Server File Structure](image)

3.3 Learner Interface Tier

The learner interface tier is where the learners interact with the treatments prepared by the instructional designers. Typically, this will occur in a web-browser. This layer of the architecture is highly customizable; generally it is defined by the ISD. The architecture supports any web-compatible format, but the learner may need to install any necessary browser software to view the training material.

4 PROTOTYPE IMPLEMENTATION

We have created a web-based system devoted to flight dispatcher icing training. This effort included content development and the creation of a flexible web-based system architecture.

4.1 Content Development

Based on our prior work (Brophy, et al, 2004), these materials were divided into three main sections: Meteorological Concepts, Weather Product Interpretation, and Icing Scenarios. Some of the content was adapted from pilot training developed by The Icing Branch at NASA Glenn Research Center (NASA, 2002). Because we plan to evaluate the usability of the system with university
students, we added specific information about the duties and operational decisions of flight dispatchers. The additional student materials will not be included in the final dispatcher training system.

In addition to developing training slides, a major effort was the development of knowledge assessments to examine the training’s effectiveness. The assessment questions were designed to map to specific areas of the training content. We adopted a pre-test, post-test methodology to apply the assessments as tools for analysis of learner improvement over the course of the training.

The first development iteration did not focus on analysis of training content, however. Our first step focused on evaluating the flexibility of our architecture and its ability to handle different training groups.

4.2 Creation of Treatments

In order to test the architecture, we mimicked a two group scenario using two training treatments. The two treatments were designed to resemble the training needs of flight dispatchers at two separate airlines (Figure 4). Treatment 1 was comprised of a meteorological concepts training component, and an assessment that related to the training. The assessment was divided into two separate components, one to be given before the training content was viewed and one to be given after the training. Treatment 2 contained a training component of basic meteorological concepts as well as an abbreviated component of the weather products interpretation section. The assessment components for this treatment were identical to the first.

4.3 Database Implementation

The system architecture as described in Section 3 was designed independently of specific database or web products. For this prototype, the database was implemented using MySQL server. Referring to the database schema shown in Figure 3, two entries were created in the treatment table (Treatment 1 and Treatment 2). Entries were created in the component (meteorology and weather products) and page tables (meteorology page 1, meteorology page 2, etc.) to associate training slides with their treatments. Entries were also created in the component (pretest and posttest) and page (pretest page 1, pretest page 2, etc.) tables to create the associated knowledge assessments.

4.4 Web-Server Implementation

The folder structure described in Figure 2 was loaded onto a web-server running Microsoft Internet Information Server (IIS). The content was loaded into the training folder in the data folder. The knowledge evaluations were loaded into the assessments folder in the data server. The server side scripts were written in PHP Hypertext PreProcessor (PHP) language and loaded into the code folder.

Figure 4: Prototype Treatments

5 FUTURE WORK

We have developed the initial architecture and prototype for the delivery of training and assessment. The next step will be to evaluate the flexibility of our architecture in a usability study using students as participants. Our goal will be to evaluate the effectiveness of the training content and assess learner performance. Then we will use actual flight dispatchers as learner participants. The dispatcher study will look at training content and provide design guidance for the final web-based flight dispatcher training system.

ACKNOWLEDGEMENTS

This work was sponsored by the Icing Branch at the NASA Glenn Research Center. Judith Van Zante and Andy Reehorst were our technical points of contact at NASA. The authors thank the personnel at the major and regional airlines that provided input to our project.
REFERENCES


AUTHOR BIOGRAPHIES

ROGER COUNIHAN is a fourth year student in the Department of Systems and Information Engineering at the University of Virginia. After graduation, he will continue his studies there to earn a Master's degree.

JUSTIN M. DEVOGE is a graduate student in the Department of Systems and Information Engineering at the University of Virginia. He received his undergraduate degree in Informatics from Indiana University. His research interests include Human Factors, Human-Computer Interaction and Instructional Systems Design.

ELLEN J. BASS is an Assistant Professor in the Department of Systems and Information Engineering at the University of Virginia. Her research interests include cognitive systems engineering, dynamic decision making, human-automation interaction, intelligent decision support systems, and intelligent learning environments.