Carnegie Mellon's PACT Lab

Data Shop v2.0
Log Analysis and Reporting

Functional Specification

Revision 2.0

August 2005

Change Log

Date	Revision	Author	Change(s)
12/10/04	1.0	Alida Skogsholm	Initial version.
4/22/05	2.0	MHCI Project Team	For version 1.1
8/2/05	2.13	MHCI Project Team	Completion of Project Course

Table of Contents

1. Overview

1.1 Introduction

The tool described in this document, the Data Shop Web Application, is intended for use by learning researchers in the Pittsburgh Science of Learning Center (PSLC). Two essential services that the Data Shop will provide are (a) a central repository of analysis data and (b) a standard reporting mechanism. The central repository of analysis data will be created from raw data collected from the seven PSLC LearnLab curricula.

Our goal is to make the analysis tool available to all participating scientists and developers working within the PSLC. We intend to cover as many standard reports as possible, allowing researchers to perform a significant amount of their analysis within the tool, but will also provide the capability to extract data so that more complex analyses can be performed in a variety of external statistical packages.

1.2 Purpose

The purpose of this document is to explain what the Data Shop Web Application and Log Conversion tools will do.

It will describe:

- Data Flow
- Necessary steps to enter log data into the analysis database
- Overall Goals
- User Scenarios
- Proposed set of Analysis Reports
- Specific Analysis Report Requirements

It will also include a glossary of terms and an appendix with the format of the info field.

1.3 Scope

This document is intended for use by developers, quality assurance, managers of the PACT group and members of the PSLC. This document uses the following terminology to specify the priority with which certain features and functionality will be addressed.

Primary – These items are determined to be feasible for implementation in the following 18 months.

Future – indicates an item that was identified in contextual inquiry and requirements solicitation as important, but was not addressed in the course of the Master's project.

1.4 Referenced Documents

- Logging and Analysis: Tools and Reports to Aid PSLC Researchers and Course Developers Project Plan
- Data Shop, Log Analysis and Reporting, Database Schema
- tutor_messge.dtd
- User research conducted by MHCI graduate students. Documentation is posted at: http://einstein.pslc.cs.cmu.edu/mhci/wiki/index.php/Main_Page

2. Data Flow

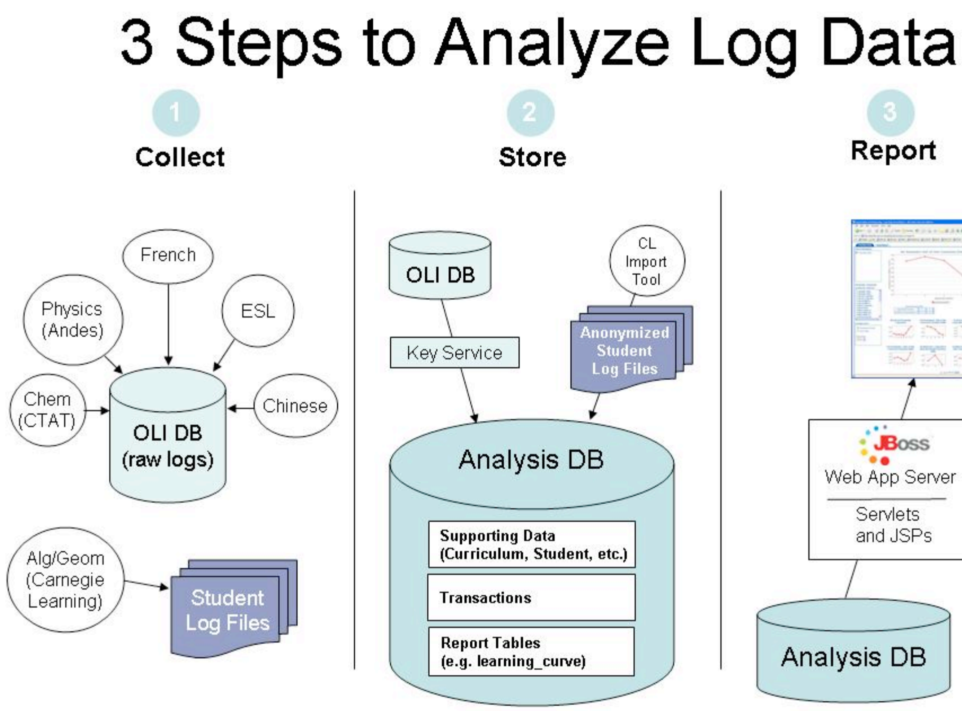


Figure 1 - Data Flow Overview

The general 3-step process depicted in the figure is as follows. First, raw log files of data will be collected as students use instrumented Learn Lab software to undertake course activities (see the left side of Figure 1). We will leverage the Open Learning Initiative's (OLI) logging capability for as much of the data collection as possible, but we will also be capable of handling pre-existing log files (e.g., Carnegie Learning's Algebra and Geometry log files). Tutor logging is expected to follow the Tutor Logging format. We will store these raw logs in a secure location. Second, the raw log files will be extracted and "cleaned" of identifying information (e.g., student names, social security numbers) and then stored in what we call an Analysis Database (middle of Figure 1). To the extent that raw logs are in pre-specified Tutor Logging format, a common extraction routine will be used. For other raw logs, such as the data from the Algebra and Geometry, specialized extraction routines will be developed. Finally, both high level and detailed reports will be made available to researchers through a web application (right side of Figure 1). Researchers will be able to select from a variety of bulk data extracts using filters such as course and time period to collect data for their own analyses offline using whatever tools they are most comfortable with. The reporting system will also allow users to browse the data with "free form" queries, guided by facilities we will provide. Finally, the reporting system will provide a gradually-increasing number of standard reports, as they are requested and specified by Learn Lab researchers.

3. Goals

3.1 Current Research Support

The web application will allow the researcher (or members of the researcher's team) to log in, view the standard reports that are appropriate to the data for their studies, drill down into statistically significant areas of the standard reports and export the data or a desired sub-set. Researchers can then work with the data in their own computing environments using Excel or another statistics package of their choice.

Additionally, an API will be provided to researchers to help them populate the Transaction table (attempt/result pairs) as well as supporting tables in a Data Shop type database. The API will allow users to write their own java programs in their own computing environments, to "distill" their raw data in whatever format it is currently in. Once their current data is in the Transaction table, researchers can perform further analyses using java, or by extracting their data from the database to do further analysis.

Contextual inquiries show that researchers have a need to quickly get a feel for the data and the stories it holds. Researchers are better supported when they can do these initial exploratory steps without interruption. "Being able to look at your results before you start doing analyses is very powerful" (CI5, 11:00) The Data Shop aims to support initial exploration via standard reports and capabilities that allow researchers to closely examine and quickly understand detailed data. Additionally, the Data Shop will make it easy for researchers to download the data as needed to perform more in-depth analyses in their own computing environments.

User research also shows that there are a few key data formats that should be initially supported. The MHCI team surveyed each of its users to help assess the most commonly used file formats to add to the user's data export tools, however no clear favorite emerged. it can be asserted that the ability to export to Excel will be practical for all users. The ability to export to a database format (SQL, mySQL) and to more complex statistical packages, primarily SPSS, will be beneficial for many users.

3.2 Additional Future Reports

As learning researchers identify reports that the entire PSLC community may find useful, the MHCI team will develop those reports into one of the web application's standard reports. These reports will become available in future releases of the web application. The current design includes a mechanism to request an additional report or provide feedback on the usefulness of the current application.

3.3 Encouraging Collaboration

Contextual inquiries also show that researchers would like to collaborate but do not actively seek new collaborators for its own sake. It is hypothesized that lowering the search costs for identifying an interesting study or identifying a researcher doing similar work will lead to collaboration opportunities.

The Data Shop Web Application design allows researchers to search and browse a virtual "table of contents," or a list of all studies. Grouped with each data set is a list of pertinent details, such as the study's principle investigator and the curricula under which it falls. Links to the papers that were published based on the particular dataset, plus necessary information to cite the dataset in a new paper are provided in the curriculum description. Finally, an access point to reports and raw data is readily visible (provided the user has the appropriate permissions).

As the data shop grows in the amount of study data it holds, the data shop will need to add a search mechanism to help researchers quickly identify relevant data sets. Here is a proposed list of meta-data that users may need most to be able to quickly identify relevant datasets:

- Principal Investigator, including contact information
- Data Custodian, including contact information
- Institution
- Geographic Location
- Demographics
- Curriculum
- Associated Papers
- Keywords

This information is accounted for in the current design, to make the eventual and inevitable conversion to a search-driven method easier as all of the necessary data will already be present for each study.

The web application will also enable users to contact the researchers or data custodians via email to ask them about particular studies or results. This will be achieved through linking to the researchers' bios on the faculty profile page of the www.learnlab.org website. PI's or data custodians will also be able to review which users have downloaded their data.

For PI's to be able to communicate the intricacies of the tutors used to create the data and for people to work collaboratively on the same data, users should be able to place global (viewed by all) annotations on datasets. (future)

3.4 Attracting New Users/Marketing (future)

The Data Shop should be a useful sales tool for recruiting new members to the PSLC. It should permit guest users to "test drive" the Data Shop capabilities on a dummy dataset. It should also provide a guided tour or tutorial for new or non- members to understand the range of features and reports available. To make it easy for a potential member to start the process of joining the PSLC, the web site should provide a form or contact information for a guest user to request to become a member.

4. User Scenarios (sample)

PSLC Primary Investigator 1 (primary)

Distinguishing User Characteristic: User comes to the Data Shop with intimate knowledge of how his tutor works and the relationship between the tutor and the logs. User is currently collecting data with his tutor for a study.

Preexisting Condition: User has a username and password for the Data Shop.

Goal: Answer some research questions that he has using the Data Shop standard reports.

1. User logs in to the Data Shop.
2. Since user is listed as the PI for the project, user has immediate access to the data for his study.
3. User accesses his dataset.
4. User decides on a question that he wants to answer with the data, and navigates to the appropriate standard report in an attempt to answer that question.
5. He notices something that looks interesting in the data (outlier, blip, spike, etc.) decides to explore it further.
6. User can see the detailed data that are contributing to the interesting phenomenon.
7. User drills in to one of the records in order to get a better idea of the story that is causing the interesting phenomenon.
8. After reviewing a few more of these records, he decides there's an interesting story and downloads all records that match the criteria.
9. Performs statistical analysis, either by downloading the data and using the statistical package and methods of his choice.

PSLC Primary Investigator 2 (primary)

Distinguishing User Characteristic: User's study has finished and she wants to know the effectiveness of the treatment in the tutor.

Preexisting Condition: User is logged into the Data Shop.

Goal: She will need to show some of the preliminary results to her research group.

1. User accesses the dataset for her experiment.
2. She browses through some information about the usage of her data set. She notices that her grad student, to whom she just granted access, downloaded the data last night. Great! He's working! (future)
3. User is interested in seeing the learning curves for each skill in the curriculum covered by her tutor to see how her tutor stacked up against the control group. She uses the Data Shop learning curve standard report and the sample selector to compare the two groups.

4. She browses the list of skill comparisons. Some of the skills she expected her tutor to teach better didn't do significantly better.
5. There are a few curves that seem to indicate that the tutor is performing better but are not statistically significant (future).
6. She finds a comparison that she wants to show her research group, so she saves it to her computer as a high-quality JPG file so she can use it in her PowerPoint presentation.

PSLC Research Programmer (primary)

Distinguishing User Characteristic: User is developing tutor and currently running a pilot test to make sure there are no hidden skills in the tutor design.

Preexisting Condition: His dataset is a pilot study; this scenario is purely a debugging session. User is currently logged in.

Goal: Make sure two skills haven't been grouped into a single opportunity.

1. User accesses the dataset for his pilot study.
2. He browses through the reports for the pilot study, and notices the extent of participation.
3. He accesses the learning curves for his data.
4. The list of curves appears. He browses through the graphs looking for a spike and finds one.
5. User accesses the details on the curve with the spike.
6. Since he's testing the skill decomposition of the problems he's written, all of the items are sequentially ordered. Attempt #6 is showing the spike.
7. He takes a closer look at Attempt #6 and determines that it's Problem #3.
8. He examines the overall performance on Problem #3 and the errors students are making on the problem.
9. The research programmer launches his tutor, visits problem #3 and thinks about how to modify the problem.

Unresolved issue: How is data from obsolete tutors deprecated? The client will address this problem later.

Student of PSLC Primary Investigator (future)

Distinguishing User Characteristic: User does not necessarily have intimate knowledge of how the tutor works and the relationship between the tutor and the logs, but she has access to an expert who does (her advisor).

Preexisting Condition: User has a username and password for the Data Shop. User has permission to view the data.

Goal: Look at a dataset following her advisor's direction.

1. User's advisor says: "we noticed before that 'x' was happening a lot, maybe you can find out why."

2. User logs in to the Data Shop.
3. Since user has examined this data several times before, it comes up on her “data recently viewed” list. (future)
4. User pulls up the dataset and decides to explore the data.
5. User formulates query to identify the records doing x (iterative).
6. When she is satisfied that she has done the correct query, she saves it for future reference.
7. User pulls up details for her first few results to begin thinking about what could be happening.
8. User downloads all records returned from the query and performs further statistical analysis on her own.

PSLC Member (no previous contact with the target data) (future)

Distinguishing User Characteristic: User is a PSLC member but has no familiarity or intimate knowledge of the tutor or the dataset.

Preexisting Condition: User has a username and password for the Data Shop. Data has been “released” to the pool of data available to the members.

Goal: User is interested in a particular phenomenon. She would like to identify whether this behavior is predictive across subjects.

1. User logs in to the Data Shop.
2. User browses the list of studies.
3. User finds one that meets her needs and decides to look into it further.
4. User formulates query to identify the records doing x (iterative).
5. When she is satisfied that she has done the correct query, she saves it for future reference and use.
6. User pulls up details for her first few results and decides that this is, indeed, what she is looking for.
7. She marks this dataset for future exportation.
8. She returns to her search results and selects another dataset.
9. She uses her saved query to search this dataset (assumes field labels same).
10. She repeats steps 6&7.
11. She repeats 8-10 until she has identified all the data she wants.
12. She exports the data that she’s identified as being useful as one merged data set.
13. The export functionality asks which statistical package she wants the data formatted for and also asks if she wants all the data as separate files or in one large file – she picks one large file.
14. She downloads the data does analysis on her own machine.

Note: this researcher is very likely to return to the Data Shop to understand anomalies she finds in the data. She will return for context, in order to answer, “How does the tutor log this question?”

Also note: Features and considerations related to inter-data set analyses were deemed out of scope for the Master’s project in a meeting with Ken Koedinger on March 23, 2005.

Non-member researcher/grad student (future)

Distinguishing User Characteristic: User is not a PSLC member and has no familiarity with the tutor.

Preexisting Condition: User does not have a username and password for the Data Shop. User does not have permission to view the data.

Goal: User would like to find and acquire a data set that may support his hypothesis.

1. User browses the dataset summaries.
2. One dataset has a paper attached that looks interesting.
3. The user opens the paper and decides to skim it. It looks interesting.
4. User closes the window and tries to download the dataset for the paper.
5. The user is prohibited from downloading the dataset, and is prompted to either log in or register with the PSLC.
6. The user goes through various business considerations (does he have to pay? How much?)
7. User fills out name, institution, reason for download, desired username and password and submits the information.
8. User receives a confirmation with the username and password as well as a direct link to the dataset she wanted to download.
9. User follows the link and logs in with her user name and password.
10. User explores data with the standard reports and then downloads the data.

5. Infrastructure

5.1 Login

5.1.1 The system must provide login functionality. (primary)

5.1.2 System must have a representation of groups. (future)

5.1.3 System must have a representation of individual researchers. (primary)

5.1.4 Users will not be permitted to save passwords to remain logged in on a local machine. (primary)

5.1.5 Sessions must time out. (primary)

5.2 Permissions

Based upon discussions with PSLC researchers and their students, it is evident that researchers expect rules to be set which govern the use of data. The Data Shop will have a built-in permissions system that enables each user to access only the data they are permitted to see.

5.2.1 The system should provide a communication channel between original data collectors (PI's) and data users. (future)

5.2.2 Each dataset should have an associated role known as its "data custodian." The person occupying this role should be able to:

5.2.2.1 *Set rules for each dataset he/she is responsible for* (future)

5.2.2.2 View a list of all datasets he/she is responsible for. (future)

5.2.2.3 Transfer the "data custodian" responsibilities to another PSLC member (future)

5.2.2.4 Grant data access to other PSLC members. (future)

5.2.3 PI's should be able to monitor the current progress of data collection. This will allow them to determine any adjustments to the tutor they may need to make, as well as to get a feel for the data during the course of the study. (primary)

5.2.4 PIs/Data custodians should be able to see who has downloaded the datasets they are responsible for (owner's watchlist). (future)

5.2.5 PIs are responsible for adding the appropriate meta-data (keywords, annotations, etc.) to their own studies. (primary)

5.3 User Interaction Logging

The system should keep track of the users' actions while using Data Shop Web Application. This is analogous to the tutors tracking the student's actions.

The system shall track the following:

5.3.1 Data set exportations (future)

5.3.2 Queries (future)

5.3.3 Data navigation (future)

5.4 Data Navigation

5.4.1 The list of studies should be scannable. Important aspects of a dataset or study (in-progress or completed, paper(s) published, etc.) should be easily identified. (primary)

5.4.1.1 When there are a sufficient number of data sets, the system should provide the ability to search based on features that researchers care about. (future)

5.4.2 Expert users should be able to run SQL queries on datasets. (future)

5.4.3 The system should provide some "turnkey" or "canned" queries. (future)

5.4.4 To enable searching, datasets must have metadata: PI, Data Custodian, Institution, Geographic Location, Demographics, Curriculum, List of associated papers, Keywords, Contact info for PIs. (primary)

5.5 Dataset Annotations (future)

5.6 Personalization

5.6.1 Users should be able to save queries in their profiles. (primary)

5.6.2 Users should be able to keep track of their "favorite" datasets. (future)

5.6.3 Users should be able to place personal annotations on datasets that cannot be viewed by others. (future)

5.6.4 The system should provide a “recently viewed data” list. This prevents users from having to repeatedly perform identical queries on the same data. (primary)

5.6.5 The system should save the users’ last view(s) upon logging out to allow researchers to continue where they left off in their analyses. (primary)

5.6.6 The system should provide an analysis history so that researchers can retrace their steps. (future)

5.6.7 Users should be able to place currently unavailable studies (i.e. studies to which they do not have access rights) on a “watch list.” (future) (variant: users should be able to contact the PI of a study to request access. (primary))

5.7 Export – Downloading (Primary)

5.7.1 Users need to be able to download datasets based on search results. (primary)

5.7.2 Users need to be able to select parameters before saving (location of file, filename, column headings y/n? etc.). (primary)

5.7.3 The system should cache export parameters so that users do not have to repeatedly specify them (primary)

5.7.4 The system should provide instructions for loading downloaded data into statistical packages. (future)

5.7.5 The system should support downloading in multiple file formats (database (SQL/mySQL), Excel, SPSS). (excel is primary)

5.7.6 Users should be able to download multiple datasets at once. (future) Note: multiple data set considerations deemed out of scope for the Master’s project in meeting on March 23, 2005.

5.7.7 If the user downloads data from more than one study, he/she needs to be able to choose to save it as one file. Each data point in this file should have a variable that indicates which study it came from. (future) Note: multiple data set considerations deemed out of scope for the Master’s project in meeting on March 23, 2005.

5.7.8 If the user downloads data from more than one study, the user needs to be able to choose to save it into separate files for each study. (future) Note: multiple data set considerations deemed out of scope for the Master's project in meeting on March 23, 2005.

6. High-Level Reports

The High-Level Reports are those that contain aggregate data.

From the "Logging and Analysis: Tools and Reports to Aid PSLC Researchers and Course Developers" project plan,

> The initial reports we propose relate to skill mastery. These reports will be valuable both to researchers studying learning and to instructors evaluating pedagogical effectiveness. Specifically, we measure learning from log entries by tracking how well students perform on successive opportunities to exercise a particular skill (Mark 1998). In an online tutor, where a student may make several attempts to answer an individual question, we determine performance by evaluating the result of the first attempt only. For instance, a student may be asked to plot a series of points on a graph: each successive point would be considered a new opportunity to exercise a "Plot-Point" skill. For each point, the first attempt is the student's first input (correct value, incorrect value or hint request) on that point. We usually consider "success" to be a correct answer (here, a point placed in the correct position on the graph) and "failure" to be either a request for help or an incorrect answer. For these reports, we order a student's first attempts at exercising a particular skill chronologically; each attempt is assigned a ordinal number from 1 to N, and these attempts are considered "opportunities to exercise a skill."

6.1 General

6.1.1 The system should provide a way to download a presentable copy of each figure and graph. (primary)

Any particular graph may be useful to the researcher for inclusion in a paper. Therefore, all graphs generated by the Data Shop web application should have the functionality to export a high-definition copy to the researcher's local machine.

6.1.2 The system should provide a way to download a spreadsheet or other file that can be used to regenerate a graphical report in another program. (primary)

The user should be able to download the numbers "behind the graph" to allow for regeneration of any figure in an external program. A researcher may want to make slight changes to a graph such as the axes names, line labels, etc. This requirement will allow such changes to be made without forcing the researcher to download the entire dataset.

6.1.3 For any report, the system should indicate the dataset contributing to the analysis. (primary)

6.2 Learning Curve

This type of curve supports analysis on the performance of a population of students on a particular skill. The X-Axis contains opportunities to exercise the skill, from 1 to N in time sequence. The Y-Axis contains one of two measures, depending on the type of graph the user prefers to generate. If the a standard learning curve is selected, the Y-axis is composed of error percentages, ranging from 0 to 100. If a latency curve is desired (see parameters below) then the Y-axis will contain measures of time ranging from 0 to the highest average time taken to successfully perform a skill on the nth attempt. Each point in a learning curve represents an average over all students in the sample. If a skill is correctly labeled in a cognitive model, a learning curve should exhibit a gradual negative slope, as the sample of students becomes more proficient (or faster, in the case of latency learning curves) at successfully performing a skill over time. If a learning curve is not monotonically downward sloping, this indicates that the underlying cognitive model has at least one instance of incorrect labeling. (Mark 1998)

6.2.1 Standard Learning Curve Graph

A learning curve indicates whether the students using the tutor are learning. A good learning curve slopes down from left to right.

6.2.1.1 The X-Axis contains opportunities to exercise the skill, from 1 to N, in time sequence. **(primary)**

6.2.1.2 The Y-Axis contains error percentages, ranging from 0 to 100. **(primary)**

6.2.2 Learning Curve Overview Screen

The learning curve overview screen shows a list of skills, each with a graph so that the researcher can quickly see which skills need his/her attention.

6.2.2.1 The user should be able to choose a curriculum. **(primary)**

6.2.2.2 The user should be able to view a list of skills by name associated with the selected curriculum. **(primary)**

6.2.2.3 The user should be able to see a small version of the standard learning curve for each skill. **(primary)**

6.2.2.4 The user should be able to navigate to the Learning Curve View Screen by clicking on an individual graph. **(primary)**

6.2.2.5 The system needs to provide the user with a way to compare curves between multiple tutors or treatments in the overview screen (2 or more tutors/treatments). This lets the user quickly identify which knowledge components require scrutiny. **(primary)**

6.2.3 Learning Curve View Screen

The learning curve view screen shows the standard learning curve graph for a particular skill and allows the researcher to pick other skills to view.

6.2.3.1 The user should be able to view the selected curriculum.

6.2.3.2 The user should be able to select a knowledge component from a list of knowledge components associated with the selected curriculum.

6.2.3.3 The user should be able to view the standard learning curve for the selected skill. (primary)

6.2.3.4 The user should be able to easily navigate to the Learning Curve Overview Screen. (primary)

6.2.3.5 The user should be able to set the number of opportunities to display. (primary)

6.2.3.6 The user should be able to exclude students who do not contribute data points in a given opportunity. (primary)

6.2.3.7 The user should be able to view the number of students for each point. (primary)

6.2.3.8 The system should provide a method of discovering which problems were presented for a particular attempt. (primary)

If attempts for the curve are from a randomized sequence of problems, the system should show the list of problems for that attempt. In order to understand an interesting point, the researcher must be able to quickly view what problem was presented to the student(s).

6.2.3.9 When a graph shows two curves, the system should show the user a measure of statistical significance. (future)

A learning curve for a skill may show that students who use one tutor/treatment are better at a particular skill than those who use a different tutor/treatment. The user also needs to know whether this difference is statistically significant.

6.2.3.10 Clearly identify whether parts of the attempts axis were presented in a randomized sequence. (future)

Attempts can be fixed sequences of problems or randomized sequences of problems.

6.3 Latency Curves: (future)

"A latency curve contains information about the behavior of a population of students on a particular skill. The nth point on the curve indicates the average time that it takes students to successfully exercise the skill on their nth first

> attempt to do so. Again, this is an indicator of learning: if the time to exercise the skill decreases across the curve, then students are believed to be learning."[1]

6.4 Help Usage Curve – (future)

Please refer to the Logging and Analysis paper for these requirements. More investigation is required to determine what will be of value to users in generating and displaying these curves.

6.4.1 Help Effectiveness Reports

[Note: This report appears be distinct from help usage curves.]
Researchers want to see the probability of errors following particular hints. [Presumably, a hint with a high probability of error needs improvement.]

This report should allow the user to compare students with the same knowledge level (based on the knowledge tracing algorithm contained in intelligent tutors). For a particular problem, researchers should be able to quickly determine if students with equal knowledge levels were better off for having taken the hint compared to students who didn't take the hint. (Did "help" help?)

6.4.1.1 The system shall support identification of help opportunities which are not helpful. (future)

6.4.1.2 For a given skill, show the help opportunities and the probability of errors after receiving a hint. (future)

6.4.1.3 The report should provide a zoomed view for each hint opportunity to show the effectiveness of the hint across several groups of students clustered by knowledge level. This view should also show the number of students in each grouping. (future)

6.5 Problem Profiles:

> "A problem profile indicates the overall performance of the group of students who completed that problem, on all skills involved in that problem. This is useful in comparing the difficulty of different problems, and in looking at the difficulty of different skills within a problem." [2]

The problem profile will display basic information for a given problem, including information specified by the tutor designer, as well as aggregate data from student performance.

---

[1] Mark, M., (1998). Analysis of Protocol Files: PACT Center User's Manual.

[2] Ibid.

6.5.1 Tutor Designer Specified Data (primary)

The problem profile will show:

- List of skills that are involved in its solution
- Number of steps from start state to solution

6.5.2 Aggregate Performance Data (primary)

The problem profile will also show, for each problem:

- The number of attempts
- The number of correct responses
- The correct answer
- The percentage of students who got this problem correct
- The number of students who asked for a hint
- The percentage of students who asked for a hint
- The most common incorrect responses and whether they were bugs or errors (moved to error reports)
- The average response time for each problem (future)

6.5.3 Multi-step Problems (primary)

Depending on the tutor design as well as the particular curriculum, some problems may have a large number of steps. All requirements in section 6.5.2 will therefore need to be provided for each *step*. Since the large amount of steps may produce a confusing display, the default setting should hide any step-specific information. The user shall be able to see the detail of a given step upon request.

6.6 Behavior Graphs (future)

A behavior graph is a profile of what a group of students did on a particular problem. It is similar to the problem profile, except that information is displayed in graphical form, and less information is immediately available.

6.6.1 The users shall be able to easily navigate from the behavior graph to the problem profile to examine a problem in detail.

6.6.2 Start State

The start state represents the problem before the students take any steps. It is represented as the top node or root of the graph. (Future development may allow the actual text of the problem to be represented in this state.)

6.6.3 Intermediate States

Other nodes branch out from the start state, each representing the resulting states of the problem after each individual student took a particular action. Each unique step that was taken from the start state has its own branch, resulting in a different node. Actions that produce states identical to those already in the graph do not receive a new node. For these actions, the count, displayed on the branch (see 6.6.5), is incremented for each additional student taking this path. (This includes nodes that were created in a higher level of the graph.) Subsequent nodes then have their own child nodes based upon student actions.

6.6.4 End States

States that do not have child nodes are considered end states. In most cases, the end state is a solution to the problem. In other cases, end states indicate the student reached a dead end and/or the tutor provided a hint intended to lead the student back to the correct path.

6.6.5 Branches

The branches, or lines that link a node to its children, will be labeled with the count of students who followed that path. The Master's students will also explore whether visual cues are helpful, (thickness of branches to indicate frequency and color of branches to indicate correct/incorrect are possible indicators). If a student's action produces a state that already exists in the graph, but was created through a different path, the graph will show the particular branch leading to this state. It is therefore possible for a graph to have loops, such that one state can have two or more states branch off from it but each of these paths can lead back to a shared state.

6.6.6 Annotations

The user shall be able to annotate all states and branches. Annotations can be useful in later analysis or to showing the behavior graph to a collaborator or graduate student. Since the actual text from the problem might not be present in these graphs, such annotations are crucial.

6.7 Error Reports (primary)

According to Mary Mark, looking for error occurrences and producing error reports serves "to gather information about the actual behavior of students who are attempting to execute production."[3] However, based on the contextual inquiries, it seems that the general purpose for error reporting is narrower: specifically, to identify which errors occurred in which contexts, and how frequently they occurred.[3]

[3] Ibid.

6.7.1 Required Error Information (Aggregate)

***6.7.1.1 The system will report how many students committed a particular error. This report will only count each student once, even if he or she committed a particular error more than once.* (primary)**

6.7.2 Sorting of Errors

***6.7.2.1 Errors will be retrievable with respect to a particular skill. The user will be able to choose whether he or she wishes to see errors made in a skill over one particular problem or all problems.* (primary)**

***6.7.2.2 Errors will also be retrievable with respect to a particular problem. In this case, the system would display all skills for that problem, and all errors made on a skill-by-skill basis in that problem.* (primary)**

6.7.3 Users should be able to distinguish between errors of commission and those of omission. (primary)

6.7.4 Clustering of Errors

As part of tutor development, errors are clustered into groups of similar wrong answers; that is, answers that are based on the same misunderstanding on the part of the student. Some tutors will have these clusters already encoded, so clusters should already be present in the database. Other cluster-related requirements include:

***6.7.4.1 The system will allow the user to create a report for all errors committed that do not fit in any predefined cluster.* (primary)**

***6.7.4.2 The system will then allow the user to specify a cluster for any non-clustered errors.* (primary)**

***6.7.4.3 Because the context of the error is so important, the user should have some way of being able to see other errors committed on the same skill or problem, even if they are already clustered.* (primary)**

***6.7.4.4 The system should allow users to annotate the addition of new clusters to provide justification to other researchers as to its creation.* (future)**

7. Detailed Reports

Detailed Reports are those that focus in on individual student data. Due to IRB constraints, these are not available to non-members of the PSLC.

7.1 General

7.1.1 The system should provide a way to download a presentable copy of each figure and graph.

Any particular graph may be useful to the researcher for inclusion in a paper. Therefore, all graphs generated by the Data Shop web application should have the functionality to export a high-definition copy to the researcher's local machine.

7.1.2 The system should provide a means to download a spreadsheet or other file with numbers that can be used to regenerate a graphical report in another program.

The user should be able to download the numbers "behind the graph" to allow for regeneration of any figure in an external program. A researcher may want to make slight changes to a graph such as the axes names, line labels, etc. This requirement will allow such changes to be made without forcing the researcher to download the entire dataset.

7.1.3 For any report, the system should indicate the dataset contributing to the analysis.

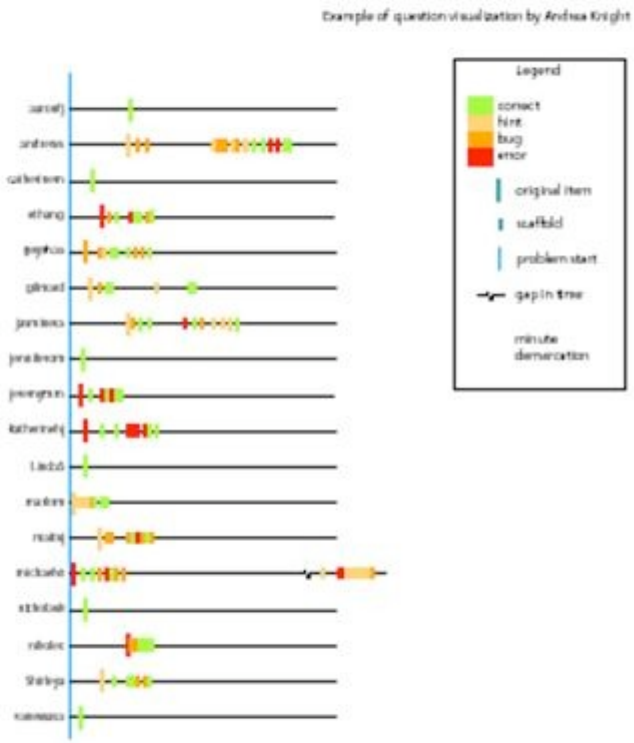
7.2 Student Learning Curve (future)

A student curve is a learning curve on an individual student. According to the logging analysis document, "the Y-Axis consists of 3 discrete points: Succeed, Help, and Fail, from the top down."

Otherwise, it supports the same functionality as the high-level learning curve.

7.3 Knight's Timeline Visualization (future)

This visualization, developed by Andrea Knight, allows the researcher to quickly see patterns in student behavior across a timeline. It can help identify which problems are difficult, where the student tended to spend more time or how often hints were used. Below is an example of this visualization. Each line represents one student. The green bar is for a correct answer, the light orange for a hint, the dark orange is for a wrong answer that was predicted and the red bar is for an incorrect answer. Each bar is placed on a timeline. The blank space between the bars indicates time where the student is reading, thinking or getting coffee.



7.4 Mostow's Session Browser (future)

7.4.1 The system should provide a session browser that allows viewing event details in a hierarchy.

7.4.2 The system should construct hierarchies which are needed by the session browser and which make sense for the users of the dataset.

7.5 Error Reports: (future)

A detailed error report reveals error details on individual students. It carries the additional requirements below. Otherwise, it supports the same functionality as the high-level error report.

7.5.1 Required Error Information (Detailed)

7.5.1.1 Given a particular error, the system will supply a list of students that committed the error.

7.5.1.2 Based on the above list of students, the system will allow researchers to look at a particular student and track his or her progress through the tutor at the time of the error (tying into* 6.6: Behavior Graphs *or* 7.3 Knight Visualizations*).

7.5.1.3 Because it is helpful to see how Help Usage may influence the occurrence of errors, users should be able to either quickly jump to a particular student's help usage or see it on the same screen as the detailed error report for a particular student.

7.6 Student Profiles: (future)

"A student profile gives an indication of a student's overall performance on all skills across all problems completed by the student. This is useful in identifying high- and low- performing students, in identifying skills which are more and less difficult for particular students, and also in identifying groups of skills which cluster together in terms of success and failure (e.g. spatial skills might cluster together relatively independent of algebraic skills on a per-student basis)."[4]

7.7 Event Filtering: (primary)

While each report should be considered a specialized data filter (that also allows for direct export), we anticipate that the most widely useful aspect of the data shop will be a filtering tool which allows the researcher to specify which data they want to analyze and how they would like the data to be displayed. Results will be presented to the researcher within the data shop and can contain summary data such as count, %, average, etc. for desired variables. If the researcher would like to do more in-depth analysis of that data set, he/she may export the data. Across all of the contextual inquiries as well as requirements solicitation, one thing was clear: there is very little consistency apparent in how learning research data analysis is conducted. And while it may be a goal of the PSLC to bring some standardization to the field, there is a prominent need for flexibility in the data shop, if only to engage those who do not use "standard reports" today.

---

[4] Mark, M., (1998). Analysis of Protocol Files: PACT Center User's Manual.

8. Log Conversion

The steps to import data into the analysis database are done manually by an administrator using the command line. There are currently two formats accepted, the OLI Log Format and the Carnegie Learning file format.

8.1 Import OLI Log Data

The OLI Log format is a tab-delimited file created by a member of the OLI group. This file will be imported into the Analysis Database manually be executing a 'LOAD DATA INFILE' command to MySQL. This file will have the following columns:

- user_id
- session_id
- source
- time
- time_zone
- action
- container
- external_obj_id
- info_type
- info

8.1.1 The Info Field Converter will extract all messages which have ' FLASH_PSEUDO_TUTOR' in the source field from the oli_log table.

8.1.2 The Info Field Converter will parse the info field as XML as specified in the tutor_message.dtd.

8.1.3 The Info Field Converter will populate the tutor_message table with these messages.

8.1.4 The Tutor Message Converter will extract messages from the tutor_message table and combine rows which go together based on the semantic_id and semantic_linked_id.

8.1.5 The Tutor Message Converter will populate the tutor_transaction table.

8.2 Import CL Log Data

The Carnegie Learning log format is well-known to those at Carnegie Learning. This format will be imported into the tutor_transaction table by a tool provided by Carnegie Learning.

8.3 Step to Skill Mapping

There will be cases where the skills are not indicated in the log data. Therefore, a tool to map steps to skills given in a file with this information will be provided. This file will be tab-delimited with the following columns:

- Problem Name
- Selection
- Action
- Input
- Skill

8.3.1 The Step to Skill Mapper will take a command line argument with the filename.

8.3.2 The Step to Skill Mapper will add problems to the problem table if the problem is not found.

8.3.3 The Step to Skill Mapper will add steps to the subgoal table if the step is not found.

Note a step is the selection, action and input values associated with a given problem.

8.3.4 The Step to Skill Mapper will add skills to the subskill table if the skill is not found.

8.3.5 The Step to Skill Mapper will associate steps to skills in the subgoal_skill_map table if the association is not found.

8.4 Learning Curve Extractor

The data needed to display a learning curve for a particular skill needs to be calculated from the data in the tutor_transaction table. This tool can be run repeatedly as more data is added to this table.

8.4.1 The Learning Curve Extractor will populate the learning_curve table.

8.4.2 The Learning Curve Extractor will calculate the total number of first attempts at a skill.

8.4.3 The Learning Curve Extractor will calculate the number of incorrect first attempts at a skill.

8.4.4 The Learning Curve Extractor will calculate the error rate by dividing the number of incorrect first attempts at a skill by the total number of first attempts at a skill.

9. Glossary

Term	Definition
CL	Carnegie Learning
Data Custodian	The primary person responsible for a dataset. This may or may not be the PI.
JBoss	The JBoss/Server is the leading Open Source, standards-compliant, J2EE based application server implemented in 100% Pure Java.
MySQL	An Open Source Software relational database management system which uses a subset of SQL.
OLI	Open Learning Initiative
PI	Primary Investigator(s) for a study. A study may have more than one PI.
Tomcat	Tomcat is an application server from the Apache Software Foundation that executes Java servlets and renders Web pages that include Java Server Page coding. Described as a "reference implementation" of the Java Servlet and the Java Server Page specifications, Tomcat is the result of an open collaboration of developers and is available from the Apache Web site in both binary and source versions. Tomcat can be used as either a standalone product with its own internal Web server or together with other Web servers.

10. Appendix A – XML Format of INFO Field

```
<?xml version="1.0" encoding="UTF-8"?>
<!--$Id: tutor_message.dtd,v 1.22 2004/12/01 22:13:40 sewall Exp $-->
<!--This DTD specifies a protocol for messages among tools and tutoring
translators and agents. The goal is to standardize the messaging format
in order to simplify the translation process for simple tutors and
pseudo-tutors and tools that want to use them. Other kinds of programs
that might use these messages include playback processors or other
instances of collaborating tools. Based on the model proposed in 'An
Architecture For Plug-In Tutor Agents' (Ritter, Koedinger 1996)-->
<!--Tutor messages may be from educational tools or from optional
tutoring software that observes user actions in the tool and may
respond to them.-->
<!--The notion of separating ui events and semantic events is taken
from the proposed IEEE LTSC specification (Ritter)-->
<!--The top-level element permits XML processors to create a well-
formed document with a single root element from a collection of tutor_
or tool_messages. No tool or tutor needs to generate a
tutor_related_message_sequence.-->
<!ELEMENT tutor_related_message_sequence (tool_message | tutor_message
| curriculum_message | message)+>
<!--The version_number attribute should be an integer that represents
the edition of this DTD to which this message set conforms. Log readers
may use this number to ensure that the reader software is compatible
with this content. -->
<!ATTLIST tutor_related_message_sequence
      version_number CDATA #IMPLIED
>
<!--A tool_message is generated by a tool and typically describes a
user action on that tool.-->
<!ELEMENT tool_message (meta?, problem_name?, (ui_event |
semantic_event)+, event_descriptor*)>
<!--An attempt_id is used when a session id is not low enough
granularity. For example, when delivered through an LMS such as OLI,
sessions may contain numerous uses of the same tool, thus the session
is not unique enough of a grouping function. Actual attempts must be
grouped together. All log entries for a particular tool use (with a
single start_tutor message) should share the same attempt_id value.-->
<!ATTLIST tool_message
      attempt_id CDATA #IMPLIED
>
<!--A tutor_message conveys a tutors evaluation of an attempt, a hint
or a scripting directive to the tool.-->
<!ELEMENT tutor_message (meta?, problem_name?, (ui_event |
semantic_event)+, event_descriptor*, action_evaluation*, tutor_advice*,
skill*, production*)>
<!--The attempt_id attribute for a tutor message has the same meaning
as that for a tool_message.-->
<!ATTLIST tutor_message
      attempt_id CDATA #IMPLIED
>
<!--A problem_name identifies the transaction context in terms of the
application. The element's structure is loosely specified:  some
applications may have a hierarchical problem-naming scheme and so would
have to specify here course, unit and section in addition to a specific
```

```
problem name. Other applications might identify problems with a simple
string or URI. The intent is to unambiguously tell what problem the
student was working on. Some messages (lock widgets, e.g.) may be
unrelated to any problem; in these cases the problem name may be
omitted.-->
<!ELEMENT problem_name ANY>
<!--The ui_event element(s) makes available a place for a tool to log
low-level user interface events. It might want a special format for
machine processing (aimed at fine-grained reproduction on playback,
e.g.) or it could be a description intended for human readers. Examples
might include a single key press, a mouse movement, etc. -->
<!ELEMENT ui_event ANY>
<!ATTLIST ui_event
      id CDATA #IMPLIED
>
<!--The semantic event is an event (at a higher level than a ui_event)
of interest to, e.g., a tutor, a higher-level playback processor or a
collaborating tool. The content of the event is unspecified in this
DTD:  it may be a human-readable or domain-specific representation of
the event. The id attribute uniquely identifies the event at least
within the session. The semantic_event_id attribute links this event to
a trigger event, so that tutor messages generated in response to tool
messages may be associated with the triggering event. The name
attribute labels the semantic event in a way significant to tutors or
other message processors:  examples of tool events include ATTEMPT,
HINT_REQUEST, START_TUTOR, STOP_TUTOR; examples of tutor events include
RESULT, HINT_MSG. The trigger attribute should be USER if the
triggering event was a user action.-->
<!ELEMENT semantic_event ANY>
<!ATTLIST semantic_event
      id ID #REQUIRED
      semantic_event_id IDREF #IMPLIED
      name CDATA #IMPLIED
      trigger (USER | DATA) #IMPLIED
>
<!--An event_descriptor contains the details of a single observable
(visible, audible, etc.) change in the state of a user interface. This
descriptor will usually detail changes initiated by users, but tutors,
e.g., could change the state of a tool's interface. In particular, for
hint responses, the event descriptor should describe the correct action
on this step: the tool may highlight the indicated selection in
conjunction with displaying the hint text.-->
<!ELEMENT event_descriptor (action*, selection*, input*)>
<!--Where multiple ui_events or semantic_events are in a message, an
event_descriptor can point to the particular XXX_event via this
reference.-->
<!ATTLIST event_descriptor
      event_id IDREF #IMPLIED
>
<!--Examples of tool actions could be ButtonPressed, UpdateTextArea,
PourSolution, etc. An example of a tutor action could be to update a
textarea or highlight a tool widget.-->
<!ELEMENT action (#PCDATA)>
<!--Actions may have unique identifiers in addition to their values-->
<!ATTLIST action
      id CDATA #IMPLIED
>
```

```
<!--A selection is the user interface widget or other element indicated
when a user takes an action. Some actions will involve multiple
selections; e.g. a PourSolution action in a lab might involve 2
selected solutions.-->
<!ELEMENT selection (#PCDATA)>
<!--Selection elements may have unique identifiers in addition to their
names.-  A selection type may distinguish the role this particular
selected item plays in an action having multiple selected items; e.g. a
PourSolution action could have a solution of type "source" and one of
type "recipient".-->
<!ATTLIST selection
      id CDATA #IMPLIED
      type CDATA #IMPLIED
>
<!--Input from a tool is generally user input, e.g., the text typed
into a textarea widget. Input from a tutor might be text that the tool
should display in a textfield.-->
<!ELEMENT input (#PCDATA)>
<!--Input elements may have unique identifiers in addition to their
values-->
<!ATTLIST input
      id CDATA #IMPLIED
>
<!--An action_evaluation is the result of a tutor's check to see
whether user input was correct. Examples of action_evaluations include
CORRECT, BUG, INCORRECT, HINT.-->
<!ELEMENT action_evaluation (#PCDATA)>
<!--An HINT action evaluation may have associated data.-->
<!ATTLIST action_evaluation
      current_hint_number CDATA #IMPLIED
      total_hints_available CDATA #IMPLIED
>
<!--The tutor_advice contains the body of a HINT_MSG, the success or
buggy message for a RESULT, etc.-->
<!ELEMENT tutor_advice (#PCDATA)>
<!--Updates to skill level(s) may be generated by a tutor as a user
completes a step in a problem. ** need an example ** -->
<!ELEMENT skill (#PCDATA)>
<!--Some skills can have an associated probability.-->
<!ATTLIST skill
      probability CDATA #IMPLIED
>
<!--A tutor might fire one or more productions to analyze a step in a
problem. An example production [from WPI]: "G.1.8-understading-polygon-
geometry".-->
<!ELEMENT production (#PCDATA)>
<!--A curriculum_message conveys information about the tutoring
context:  it may describe the exercise in its global context of school,
course, etc. It may include information that a curriculum management
system needs to select a next exercise.-->
<!ELEMENT curriculum_message (meta?, school_name?, course_name?,
unit_name?, section_name?, problem_name?, dfa*, skill*, production*)>
<!--The attempt_id attribute for a curriculum message has the same
meaning as that for a tool_message. The name attribute gives the
message purpose; examples include START_PROBLEM, STOP_PROBLEM,
UPDATE_SKILL_MASTERY.-->
<!ATTLIST curriculum_message
```

```
      attempt_id CDATA #IMPLIED
      name CDATA #IMPLIED
>
<!--An school is the institution or other organization where this
activity is taking place.-->
<!ELEMENT school_name (#PCDATA)>
<!--A course is a course of study, such as 8th-grade mathematics or
microeconomics or intermediate French, or the title of a textbook.-->
<!ELEMENT course_name (#PCDATA)>
<!--A unit is a high-level division of a course.-->
<!ELEMENT unit_name (#PCDATA)>
<!--A section is a lower-level division of a course.-->
<!ELEMENT section_name (#PCDATA)>
<!--Difficulty factors analysis is data associated with this problem.
We recommend data of this sort, if available, be included with
information logged by a START_PROBLEM entry.-->
<!ELEMENT dfa ANY>
<!--The message element is a facility for arbitrary communications
content to be accommodated in this DTD. A message might contain one or
more property elements.-->
<!ELEMENT message ANY>
<!--A property is a name-value pair. The value might be a list of
entries.-->
<!ELEMENT property (#PCDATA|property)*>
<!ATTLIST property
      name CDATA #REQUIRED
>
<!--An entry is a single value in a list of entries.-->
<!ELEMENT entry (#PCDATA)>
<!--Meta level data. This block of data may be known to the agents,
translators or tools without needing it to be encoded in the message
itself. However, if such data is not available, it can be encoded in
the message, although it is really part of the communication. Given
potential differences in location of agents and tools (often
experienced with a web-based tool), time zone is required.-->
<!ELEMENT meta (user_id, time, time_zone, session_id)>
<!ELEMENT user_id (#PCDATA)>
<!ELEMENT time (#PCDATA)>
<!ELEMENT time_zone (#PCDATA)>
<!ELEMENT session_id (#PCDATA)>
```