An Open Repository and analysis tools for fine-grained, longitudinal learner data.

Conference Paper · January 2008

Source: DBLP

4 authors, including:

Kenneth R. Koedinger
Carnegie Mellon University
386 PUBLICATIONS **13,917** CITATIONS

Alida Skogsholm
Carnegie Mellon University
8 PUBLICATIONS **278** CITATIONS

Brett Leber
Carnegie Mellon University
8 PUBLICATIONS **268** CITATIONS

An open repository and analysis tools for fine-grained, longitudinal learner data

Kenneth R. Koedinger, Kyle Cunningham, Alida Skogsholm, and Brett Leber
{krk, kcunning, alida, bleber}@cs.cmu.edu
Human Computer Interaction Institute, Carnegie Mellon University

Abstract. We introduce an open data repository and set of associated visualization and analysis tools. The Pittsburgh Science of Learning Center's "DataShop" has data from thousands of students deriving from interactions with on-line course materials and intelligent tutoring systems. The data is fine-grained, with student actions recorded roughly every 20 seconds, and it is longitudinal, spanning semester or yearlong courses. Currently over 110 datasets are stored including nearly 18 million student actions. Most student actions are "coded" meaning they are not only graded as correct or incorrect, but are categorized in terms of the hypothesized competencies or knowledge components neede to perform that action. Researchers have analyzed these data to better understand student cognitive and affective states and the results have been used to redesign instruction and demonstrably improve student learning.

1 Introduction

One reason for emerging excitement about educational data mining is the increasing availability of student learning data. These data are coming from many sources including data from schools, like standardized tests and student and teacher background variables (e.g., www.icpsr.umich.edu/IAED), and videos of classroom interactions (e.g., www.talkbank.org). In this paper we present an open data repository of learning data coming primarily from computer "click stream" data that arises from student interaction and system response in online courses, online assessment, intelligent tutoring systems, virtual labs, simulations, and other forms of educational technology.

Extensive new data sources have been transformational in science [2] and business (e.g., Google, FedEx). Learning scientists can also benefit from having large repositories of data available, easily accessible, and with associated data visualization and analysis tools. Moving toward a common set of standards for storing data will facilitate more efficient, extensive storage and use of such data, as well as greater sharing of visualization and analysis techniques. Providing learning scientists with rich student learning data and advancing toward common standards are key goals of the Pittsburgh Science of Learning Center's DataShop (http:// learnlab.org/datashop).

2 The Pittsburgh Science of Learning Center's *DataShop*

DataShop is a data repository and web application for learning science researchers. It provides secure data storage as well as an array of analysis and visualization tools available through a web-based interface. Primarily, DataShop stores learner interactions from online course materials that include intelligent tutors. Data is collected from the seven PSLC courses: Algebra, Chemistry, Chinese, English, French, Geometry and

Physics. There are also sources external to the PSLC that contribute to DataShop, such as middle school math data from the Assistments project (http://www.assistment.org).

DataShop can store any type of data associated with a course or study. This includes intelligent tutor data (which is capable of being analyzed through the analysis and visualization tools) as well as any related publications, files, presentations, or electronic artifacts a researcher would like to store. Courses and studies are represented as datasets, which are organized by project. An example of this is the "Algebra 1 2005-2006" dataset, which is grouped with similar datasets under the "Algebra Course" project. Semantic information can be stored for each dataset, providing additional context.

The amount of data in DataShop is constantly growing. As of May 2008, DataShop offers 114 datasets under 42 projects, most generated within the past 4 years. The majority of datasets contain tutor interaction data, while others are files-only datasets, used solely for central and secure file storage. There are about 18 million tutor transactions, representing about 70,000 student hours available for analysis.

Researchers have utilized DataShop to explore learning issues in a variety of educational domains. These include, but are not limited to, collaborative problem solving in Algebra [17], self-explanation in Physics [9], the effectiveness of worked examples and polite language in a Stoichiometry tutor [11] and the optimization of knowledge component learning in Chinese [14].

2.1 Logging & Storage Models

Tutor data is typically parsed from messages logged by automated computer tutors employing the DataShop XML format, such as the one depicted in Figure 1.

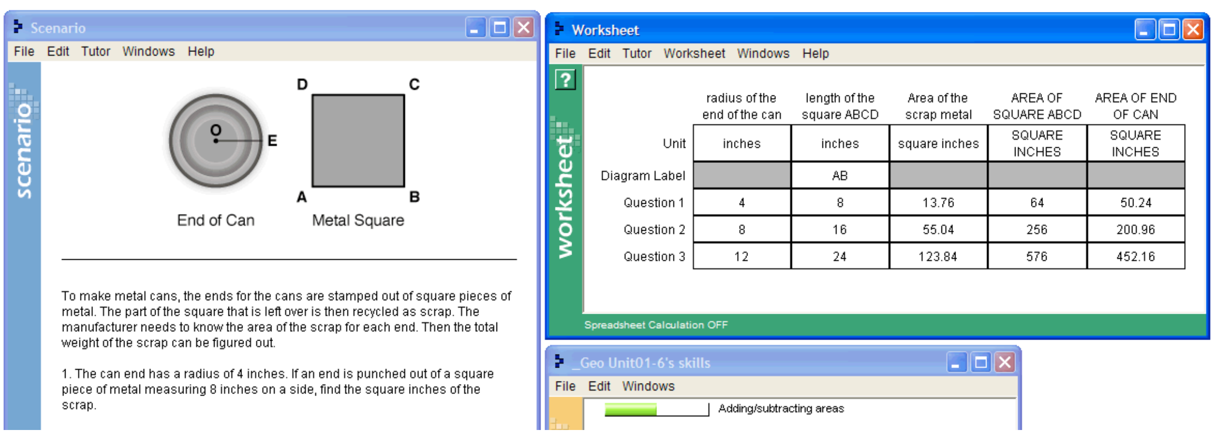

Figure 1. An example problem, "Making Cans", from Carnegie Learning's Cognitive Tutor 2005.

In the "Making Cans" example shown in Figure 1, a student completes a problem where she is asked to find the area of a piece of scrap metal left over after removing a circular area (the end of a can) from a metal square. The student enters everything in the worksheet except for the row labels, and column and 'Unit' labels for the first three columns. In addition to providing students with feedback and instruction, the tutor records the student's actions and tutor responses, and stores them in a log file, which is imported and analyzed by DataShop.

The DataShop logging format differs from existing standard formats in that it attempts to capture student-tutor interaction history at a fine-grained level, while also providing context for such interactions; the format does not attempt to describe, *a priori*, learning resources and how they're transferred (e.g., LOM, SCORM) or test content (e.g., IMS-QTI). In this way, the format is essentially descriptive, not prescriptive. The DataShop logging model is represented by the following constructs [16]:

- Context message: the student, problem, and session with the tutor
- Tool message: represents an action in the tool performed by a student or tutor
- Tutor message: represents a tutor's response to a student action

Below we see example context, tool, and tutor messages in the DataShop XML format:

```
<context_message context_message_id="C2badca9c5c:-7fe5" name="START_PROBLEM">
  <dataset> <name>Geometry Hampton 2005-2006</name>
    <level type="Lesson"> <name>PACT-AREA</name>
      <level type="Section"> <name>PACT-AREA-6</name>
        <problem> <name>MAKING-CANS</name> </problem>
      </level>
    </level>
  </dataset>
</context_message>
<tool_message context_message_id="C2badca9c5c:-7fe5">
    <semantic_event transaction_id="T2a9c5c:-7fe7" name="ATTEMPT" />
    <event_descriptor>
      <selection>(POG-AREA QUESTION2)</selection>
      <action>INPUT-CELL-VALUE</action>
      <input>200.96</input>
    </event_descriptor>
  </tool_message>
<tutor_message context_message_id="C2badca9c5c:-7fe5">
    <semantic_event transaction_id="T2a9c5c:-7fe7" name="RESULT" />
    <event_descriptor> … [as above] … </event_descriptor>
    <action_evaluation>CORRECT</action_evaluation>
  </tutor_message>
```

In this example, the student attempted problem "MAKING-CANS" in the "PACT-AREA" lesson of the Geometry tutor. Looking at the tool and tutor message pair, we see the student correctly entered "200.96" as the answer. Tool and tutor messages are traditionally paired with each other (by context message), allowing DataShop to interpret the student action and the tutor's response in conjunction. These pairs are then stored as a single *tutor transaction* in the database. Table 1 below illustrates how actions from the Making Cans example are interpreted and stored as tutor transactions.

A tutor transaction stores details such as the student(s), session, time, problem, problem subgoal (step), attempt number, student input, tutor response, number of hints, conditions assigned to the problem step, as well as knowledge components (skills). Unpaired tool messages can be used to represent untutored actions, such as a student starting audio playback, and are stored in the repository as well.

Table 1. A simplified tutor transaction excerpt from the "Making Cans" example.

#	Student	Problem	Step	Attempt #	Student Input	Evaluation	Knowledge component
1	S01	MAKING-CANS	(SQUARE-BASE Q1)	1	8	CORRECT	Enter-Given
2	S01	MAKING-CANS	(SCRAP-METAL-AREA Q1)	1	32	INCORRECT	
3	S01	MAKING-CANS	(SCRAP-METAL-AREA Q1)	2	4	INCORRECT	
4	S01	MAKING-CANS	(SQUARE-AREA Q1)	1	64	CORRECT	Square-Area
5	S01	MAKING-CANS	(POG-AREA Q1)	1	50.24	CORRECT	Circle-Area
6	S01	MAKING-CANS	(SCRAP-METAL-AREA Q1)	3	13.76	CORRECT	Compose-Areas
7	S01	MAKING-CANS	(POG-RADIUS Q2)	1	8	CORRECT	Enter-Given
8	S01	MAKING-CANS	(SQUARE-BASE Q2)	1	16	CORRECT	Enter-Given
9	S01	MAKING-CANS	(SQUARE-AREA Q2)	1	256	CORRECT	Square-Area
10	S01	MAKING-CANS	(POG-AREA Q2)	1	200.96	CORRECT	Circle-Area
11	S01	MAKING-CANS	(SCRAP-METAL-AREA Q2)	1	55.04	CORRECT	Compose-Areas
12	S01	MAKING-CANS	(POG-RADIUS Q3)	1	12	CORRECT	Enter-Given
13	S01	MAKING-CANS	(SQUARE-BASE Q3)	1	24	CORRECT	Enter-Given
14	S01	MAKING-CANS	(SQUARE-AREA Q3)	1	576	CORRECT	Square-Area
15	S01	MAKING-CANS	(POG-AREA Q3)	1	452.16	CORRECT	Circle-Area
16	S01	MAKING-CANS	(SCRAP-METAL-AREA Q3)	1	123.84	CORRECT	Compose-Areas
17	S01	MAKING-CANS	DONE	1	DONE	CORRECT	Determine-Done

Multiple tool and tutor messages are typically logged for a single problem-solving activity. Problem-solving activity is broken down into "steps" which represent completion of possible subgoals or pieces of a problem solution. Students often make multiple attempts at a step or get instructional help on a step and each of these attempts or help requests are stored as a separate tutor transaction in the database.

In the "Making Cans" example, we see the student attempted the "(SCRAP-METAL-AREA Q1)" step three times (transaction numbers 2, 3 and 6 in Table 1). We can ascertain from the transactions that the student was unsuccessful in her first two attempts, providing an answer of "32" and "4", both labeled as incorrect by the tutor. On the third attempt, the student successfully completed the problem step, providing an input of 13.76 (as can be seen in Figure 1).

To allow for fast and easy visualization and analysis of data, tutor transactions are aggregated into a student-step rollup table. This "denormalized" table aggregates steps by student, problem, and step and is used by many of the DataShop tools, such as the Performance Profiler and Learning Curve. An example of how the "Making Cans" tutor transactions are aggregated by student-step is depicted in Table 2.

The student-step roll-up table stores an array of useful step information, some of which is depicted above. In the "(SCRAP-METAL-AREA Q1)" step in the "Making Cans" example, we can see that the three tutor transactions are now rolled into a single step entry (line number 9 in Table 2). "Opportunity count" is two since the student saw the "Compose-Areas" knowledge component twice. "Total incorrects" is two, since she made two incorrect attempts. “Assistance score”, which consists of total hints added to total incorrects, is two. “Error rate”, whether the first attempt was an incorrect action or hint request, is 1 (or 100%) because the student’s first attempt was an error.

Table 2. Data from the “Making Cans” example, aggregated by student-step

Step table excerpt

#	Student	Problem	Step	Opportunity Count	Total Incorrects	Total Hints	Assistance Score	Error Rate	Knowledge component
1	S01	WATERING_VEGGIES	(WATERED-AREA Q1)	1	0	0	0	0	Circle-Area
2	S01	WATERING_VEGGIES	(TOTAL-GARDEN Q1)	1	2	1	3	1	Rectangle-Area
3	S01	WATERING_VEGGIES	(UNWATERED-AREA Q1)	1	0	0	0	0	Compose-Areas
4	S01	WATERING_VEGGIES	DONE	1	0	0	0	0	Determine-Done
5	S01	MAKING-CANS	(POG-RADIUS Q1)	1	0	0	0	0	Enter-Given
6	S01	MAKING-CANS	(SQUARE-BASE Q1)	1	0	0	0	0	Enter-Given
7	S01	MAKING-CANS	(SQUARE-AREA Q1)	1	0	0	0	0	Square-Area
8	S01	MAKING-CANS	(POG-AREA Q1)	2	0	0	0	0	Circle-Area
9	S01	MAKING-CANS	(SCRAP-METAL-AREA Q1)	2	2	0	2	1	Compose-Areas
10	S01	MAKING-CANS	(POG-RADIUS Q2)	2	0	0	0	0	Enter-Given
11	S01	MAKING-CANS	(SQUARE-BASE Q2)	2	0	0	0	0	Enter-Given
12	S01	MAKING-CANS	(SQUARE-AREA Q2)	2	0	0	0	0	Square-Area
13	S01	MAKING-CANS	(POG-AREA Q2)	3	0	0	0	0	Circle-Area
14	S01	MAKING-CANS	(SCRAP-METAL-AREA Q2)	3	0	0	0	0	Compose-Areas
15	S01	MAKING-CANS	(POG-RADIUS Q3)	3	0	0	0	0	Enter-Given
16	S01	MAKING-CANS	(SQUARE-BASE Q3)	3	0	0	0	0	Enter-Given
17	S01	MAKING-CANS	(SQUARE-AREA Q3)	3	0	0	0	0	Square-Area
18	S01	MAKING-CANS	(POG-AREA Q3)	4	0	0	0	0	Circle-Area
19	S01	MAKING-CANS	(SCRAP-METAL-AREA Q3)	4	0	0	0	0	Compose-Areas
20	S01	MAKING-CANS	DONE	2	0	0	0	0	Determine-Done

Each step in a problem requires the student to know something—a relevant concept or skill—to perform the step correctly. This small unit of knowledge is termed a “knowledge component”, a key notion in the PSLC’s theoretical framework (learnlab.org/research/wiki). To document this required concept or skill, a tutor author labels steps with a hypothesized knowledge component(s) required for correct completion of the step. In the “Making Cans” example, we see the knowledge component “Compose-Areas” assigned to the correct transaction (row 6 of Table 1) for the “(SCRAP-METAL-AREA Q1)” step.

A knowledge component codes for a general student capability to accomplish steps in tasks. Knowledge component modeling, the process of assigning knowledge components to steps, bolsters the usefulness of intelligent tutor data by providing additional context for a step. A step can have zero, one or multiple knowledge components associated with it. Steps that have assigned knowledge components are used to produce a variety of “learning curves”. Figure 2 below depicts error rate learning curves generated by DataShop. In this graph, error rate, or the percentage of students that asked for a hint or made an incorrect attempt on their first attempt on steps associated with the knowledge component, is shown on the y-axis. The x-axis (“Opportunity”) indicates the nth time (e.g., 4 is the 4th time) a student has (in theory) used (or tutored on) a knowledge component to solve a step in a problem. Each unique step in a problem is distinct from other problem-solving steps. For example, in Table 2, “(POG-AREA Q1)” and “(POG-AREA Q2)” are unique steps—they correspond to different questions in problem. Because they both exercise the “Circle-Area” knowledge component, they are counted as distinct opportunities for the student to demonstrate whether he or she has learned "Circle-Area" (and if not, get instruction on it).

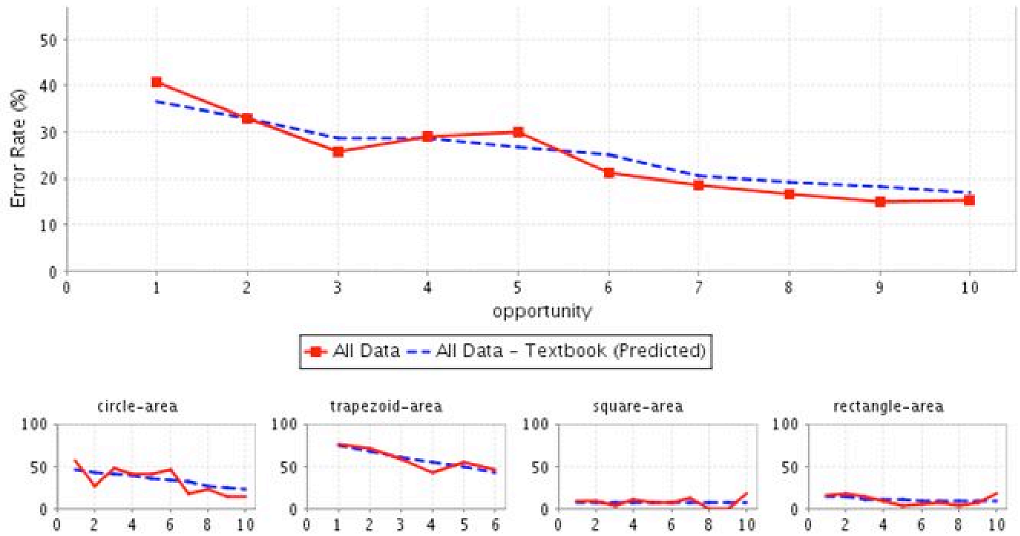

Figure 2. Error Rate Learning Curve with predicted values from a Geometry Area dataset.

The solid curve represents the actual values, each point an average across all students and knowledge components for the given opportunity. The dashed curve represents the predicted curve values, based on the Learning Factor Analysis (LFA) algorithm [5]. Built-in statistical models measure student proficiency, knowledge component difficulty, and knowledge component learning rates. This algorithm allows for search and comparison of alternative knowledge component or cognitive models. DataShop supports users in entering multiple hypothesizes about how student knowledge and learning may be decomposed.

2.2 Importing and Exporting Learning Data

Data may be imported into the DataShop repository through XML or a tab-delimited text file format. Logging to DataShop XML provides the richest and most complete data. If logging via XML, tutors can send messages directly to the DataShop logging server in real time. Logs are automatically processed on a nightly basis, making them available for analysis or export through the web application. Alternatively, a computer tutor can write XML to files on the local hard disk (for example, if the tutor is running off-line) and then send the data to the logging server at a later time. Data in a pre-existing log format can also be converted to DataShop XML and then imported into the repository. This procedure has worked well for data collected by other tutoring systems including Andes (www.andes.pitt.edu), math Cognitive Tutors (carnegielearning.com), REAP (reap.cs.cmu.edu), and Assistments (assistment.org). The tab-delimited format of a transaction table can alternatively be used to import from a preexisting source.

DataShop offers various data export options through the web application each delivered in a tab-delimited text file. These include transaction and student-step level exports (as illustrated in Tables 1 & 2), and a student-problem aggregate export.

2.3 Analysis & Visualization Tools

The DataShop web application provides several tools to assist with analyzing and visualizing repository data. These tools can be used in conjunction to jump-start data analysis: one can determine if students are learning by viewing learning curves, then drill down on individual problems, knowledge components, and students to analyze performance measures.

The following DataShop tools are available:

- *Dataset Info:* provides dataset metrics, contextual information, quick statistics (number of students, transactions, knowledge components, etc.) as well as papers, files, a problem table, and exporting and importing knowledge component models.
- *Performance Profiler:* multi-purpose tool that visualizes student performance across various dataset-specific domains (problem, step, curriculum level, knowledge component and student) and measures of performance (error rate, assistance score, average number of incorrects, average number of hints, and residual error rate).
- *Error Report:* presents each student's first attempt at a problem or knowledge component, including if he or she was correct, the number of students or observations, and the text of the student's answer.
- *Learning Curve:* visualizes student learning changes over time. Learning curve types include error rate, assistance score, correct step time, and others. The Learning Factors Analysis model [6] provides predicted values for error rate learning curves.

3 Uses of PSLC's DataShop

As indicated above, many recent analyses of data from DataShop have been performed in a variety of domains. A number of other studies have used, tested or extended the analysis techniques employed in DataShop including investigations in reading [10], Physics [12], and Geometry [15]. Often analyses have been targeted at finding ways to improve student learning. In some cases, the work has been taken full circle such that an analysis led to an instructional redesign that was demonstrated to improve student learning beyond that realized by the original instruction. We provide a couple examples.

Cen, Junker, and Koedinger [6] performed a learning curve analysis using the Learning Factors Analysis (LFA) algorithm based on data from the Area unit of the Geometry Cognitive Tutor. They noticed that while students were required to over-practice some easy target knowledge components or skills (see square-area in Figure 2), they under-practiced some harder skills (see trapezoid-area in Figure 2). Based on observation and further analysis, they created a new version of the geometry tutor by resetting parameters that determine how often skills are practiced. They ran a classroom experiment where students in a course were pre- and post-tested and randomly assigned to use either the previous or the new tutor version. Students using the new version took 20% less time to finish the same curriculum units (because over-practice was eliminated) and learned just as much as measured by normal, transfer, and long-term retention tests.

A second demonstration of a datamining project that "closed the loop" is work by Baker et al. [3], who had done formal observations of student behavior in computer labs while working through lessons of a middle school math Cognitive Tutor. Among a number of categories of off-task or otherwise disengaged behavior, he found that "gaming the system" had the largest correlation with poor learning outcomes. Gaming refers to student behavior that appears to avoid thinking and learning through systematic guessing or fast and repeated requests for increasing help. Baker used machine learning techniques to build a "detector" capable of processing student log information, in real time, to determine when students were gaming. The detector became the basis for an intervention system, a "meta tutor", designed to discourage gaming and engage students in supplementary instruction on topics they had gamed. A controlled experiment demonstrated student-learning benefits associated with this adaptive selection of supplementary instruction for students observed to be gaming.

4 Discovering better cognitive and affective models of student learning

An important general use of this kind of data is to drive the development of more precise computational models of human cognition, motivation, and learning. The work on gaming is just one example of using interaction log data to assess and remediate student motivation and other detectors of student engagement are possible.

With respect to cognitive and learning assessment, we have been pursuing a strong hypothesis that the correct representation of knowledge (facts, skills, concepts, strategies, integrative models or schemas, meta-knowledge, etc.) in a domain is an empirical question. Why is this a strong claim? First, many do not believe knowledge is decomposable, for instance, Barab and Squire say "learning, cognition, knowing, and context are irreducibly co-constituted and cannot be treated as isolated entities or processes" [4]. Second, most designers of instruction, like textbook authors, assume that the instructional designers determine the knowledge decomposition, the target concepts and skills, and students learn these knowledge components as they are taught. In contrast, the PSLC takes the position that the nature and grain size of mental knowledge representation is driven as much or more by the student and the explicit and implicit learning mechanisms in which they and their brains engage. Others, too, have emphasized that students do not learn knowledge in pieces as they are taught [7]. If the nature of human knowledge representation is an empirical question, then we need both vast and detailed educational data and associated data processing algorithms to answer this question for all domains of academic learning.

In addition to the LFA technique mentioned above, which employs the "smooth learning curve" criteria [13], a number of algorithms have been created for empirically evaluating knowledge representations against student performance data including the rule space [18] and knowledge space [8] approaches. The collection and use of on-line learning data will further drive such developments. As noted in a major national report "psychometric validation of [on-line] assessments is needed so they can be compared with conventional assessments, and complement and ultimately supplant them" [1].

5 Conclusion

We described PSLC's DataShop, an open repository and web-based tool suite for storing and analyzing click-stream data, fine-grained longitudinal data generated by online courses, assessments, intelligent tutoring systems, virtual labs, simulations, and other forms of educational technology. In contrast to other types of educational data such as video and school-level data, data in DataShop includes a rich set of semantic codes that facilitate automated analysis and meaningful interpretation.

The PSLC DataShop uniform data format is an initial attempt to develop a common standard that we hope will be useful to field if not as is, then in driving better or more useful common standards. In addition to being a source for learning data, it is also a place where researchers can deposit data and then get help from other researchers who can perform secondary analysis on this data.

DataShop allows free access to a wide variety of data sets and analysis tools. These tools help researchers visualize student performance, difficulties, and learning over time. Such analyses can lead to demonstrably better instructional designs. The data can also drive improved models of student cognition, affect, and learning that can be used to improve on-line assessment and on-line learning. We take as a premise that the human brain constructs knowledge based a variety of input sources (e.g., verbal, visual, physical) and in a fashion and at a grain size that may or may not conform to the structure as conceived by an instructor or domain expert. In other words, the nature and content of human knowledge representation is a deep and important scientific question, like for instance, the nature of the human genome. To answer this question requires a vast collection of relevant data, associated analysis methods and new theory.

Acknowledgement

Research supported by the National Science Foundation award number SBE-0354420.

References

1. Ainsworth, S., Honey, M., Johnson, W. L., Koedinger, K. R., Muramatsu, B., Pea, R., Recker, M., & Weimar, S. Cyberinfrastructure for Education and Learning for the Future (CELF): A Vision and Research Agenda, 2005. Washington, DC: Computing Research Association. http://www.cra.org/reports/cyberinfrastructure.pdf.
2. Atkins, D.E. (Ed.). Revolutionizing Science and Engineering Through Cyberinfrastructure: Report on the National Science Foundation Blue-Ribbon Advisory Panel on Cyberinfrastructure, 2003. Arlington, Virginia: National Science Foundation. http://www.cise.nsf.gov/sci/reports/atkins.pdf
3. Baker, R., Corbett, A., Koedinger, K. R., Evenson, S., Roll, I., Wagner, A., Naim, M., Raspat, J., Baker, D., & Beck, J. Adapting to when students game an intelligent tutoring system. In M. Ikeda, K. D. Ashley, T.-W. Chan (Eds.) *Proceedings of the 8th International Conference on Intelligent Tutoring Systems,* 2006, p. 392-401. Berlin: Springer-Verlag.

4. Barab, S. & Squire, K. Design-based research: Putting a stake in the ground. *The Journal of the Learning Sciences,* 2004, 13(1).
5. Cen, H., Koedinger, K., & Junker, B. Learning Factors Analysis - A general method for cognitive model evaluation and improvement. *8th International Conference on Intelligent Tutoring Systems,* 2006.
6. Cen, H., Koedinger, K., & Junker, B. Is over practice necessary? – Improving learning efficiency with the cognitive tutor through educational data mining. In Rose Luckin and Ken Koedinger (Eds.) *Proceedings of the 13th International Conference on Artificial Intelligence in Education*, 2007. p. 511-518. Los Angeles: IOS Press.
7. diSessa, A. A. Toward an epistemology of physics. *Cognition and Instruction,* 1993, 10(2 & 3), p. 105-225.
8. Falmagne, J.-C., Koppen, M., Villano, M., Doignon, J.-P., et al. Introduction to knowledge spaces: How to build, test, and search them. *Psychological Review,* 1990, 97, p. 201-224.
9. Hausmann, R., & VanLehn, K. Self-explaining in the classroom: Learning curve evidence. In McNamara & Trafton (Eds.), *Proceedings of the 29th Annual Cognitive Science Society*, 2007, p. 1067-1072. Austin, TX: Cognitive Science Society.
10. Leszczenski, J. M. & Beck J. E. What's in a word? Extending learning factors analysis to model reading transfer. *Proceedings of the Educational Data Mining workshop* held at the 14th International Conference on Artificial Intelligence in Education, Los Angeles, 2007, p. 31-39.
11. McLaren, B. M., Lim, S., Yaron, D., & Koedinger, K. R. Can a polite intelligent tutoring system lead to improved learning outside of the lab? In Luckin & Koedinger (Eds.) *Proceedings of the 13th International Conference on Artificial Intelligence in Education*, 2007, p. 433-440. Los Angeles: IOS Press.
12. Nwaigwe, A., Koedinger, K.R., VanLehn, K., Hausmann, R., & Weinstein, A. Exploring alternative methods for error attribution in learning curves analyses in intelligent tutoring systems. In Luckin & Koedinger (Eds.) *Proceedings of the 13th International Conference on Artificial Intelligence in Education*, 2007. p. 246-253.
13. Ohlsson, S. & Mitrovic, A. Constraint-based knowledge representation for individualized instruction. *Computer Science and Information Systems*, 2006, 1-22.
14. Pavlik Jr., P. I., Presson, N., & Koedinger, K. R. Optimizing knowledge component learning using a dynamic structural model of practice. In R. Lewis & T. Polk (Eds.) *Proceedings of the Eighth International Conference of Cognitive Modeling*, 2007.
15. Rafferty, A. N. & Yudelson, M. Applying learning factors analysis to build stereotypic student models. *Proceedings of 13th International Conference on Artificial Intelligence in Education,* 2007.
16. Ritter, S., & Koedinger, K. R. An architecture for plug-in tutor agents. *Journal of Artificial Intelligence in Education,*1996, 7(3-4), p. 315-347.
17. Rummel, N., Spada, H., & Diziol, D. Evaluating collaborative extensions to the Cognitive Tutor Algebra in an in vivo experiment. Lessons learned. Paper presented at the 12th European Conference for Research on Learning and Instruction (EARLI) 2007. Budapest, Hungary.
18. Tatsuoka, K. Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement,* 1983, 20(4), p. 345-354.