

Mental Models of Data

Leigh Ann Sudol
Computer Science
Department
Carnegie Mellon University
Pittsburgh, PA
lsudol@andrew.cmu.edu

Mark Stehlik
Computer Science
Department
Carnegie Mellon University
Pittsburgh, PA
mjs@cs.cmu.edu

Sharon Carver
Psychology Department
Carnegie Mellon University
Pittsburgh, PA
sc0e@andrew.cmu.edu

ABSTRACT

This discussion paper describes the results of a pilot study into the mental models of data and data structures held by people based upon the software applications that they use frequently. Computers and data intensive software like email and iTunes have become a large part of our daily lives. The results are provided to motivate discussion about prevalent models of data and potential impact that they may have on introductory curriculum design. Results suggest that people tend to hold abstract models of the data types and regardless of gender or application being discussed there is a common structure applied by people.

Keywords

Mental Models, Introductory Programming, Computer Science Education

1. INTRODUCTION

Over the past 40 years introductory computer science education has favored making curriculum changes based on external demands such as popular programming languages or paradigms. This has introduced additional complexity into our courses, especially early on. In this time students have come to computer science class with increasing experience in using computers and computer software. The Common Sense Computing series of papers have sought to explore the naive concepts that students come to class with [8, 7].

As students have changed in their pre-existing knowledge over the last 40 years, so has the software that they work with on a daily basis. With an increase in social networking, communication and data storage and organization on the computer, the software applications in these domains have also become more sophisticated. Through repeated use of these applications, the current generation of students have developed sophisticated models of the information that drives the software as well as the interaction with this information.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Koli Calling '09 October 29 - November 1, 2009, Koli, Finland
Copyright 2009 ACM 978-1-60558-952-7/09/11 ...\$10.00.

In this paper I explore the results of a pilot study aimed at assessing the way that people express the models that they have for working with this information. I discuss the methodology of data collection, the coding of the interviews, and the results. While not generalizable because of the small sample size, the results do point to a pre-existing model of data that is consistent across many applications and is based more in abstract representations and the interaction provided by the software application to the data. Finally, open questions prompted by this research are proposed for discussion.

2. MENTAL MODELS IN EDUCATION

The process of learning involves the connection between existing knowledge and new material. Without providing that connection, meaningful learning does not occur [5]. These two particular components of learning are equally important to be addressed, and yet often prior knowledge is not activated or assumed not to exist when instruction is going on. Many studies in cognitive science and conceptual change have shown us that attention must be paid to the knowledge and beliefs that students have prior to instruction[2].

A mental model is an internal representation of external objects or events[3]. They deal with both the form or structure of the item as well as the rules for interaction with the item[12]. Today's students have formed mental models for the data intensive applications that they work with on a daily or weekly basis. It is up to us as educators to take advantage of these models in our instruction.

With current instructional practices, students are not exiting introductory courses with appropriate models for computer science[4]. Without correctly identifying the models that students already hold before trying to impose new models, it is difficult to know if information is being lost in translation.

Cognitive science tells us that utilizing pre-existing models can help reduce cognitive load of students and help teachers find the zone of proximal development that is appropriate for their students[2, 10, 11]. Studies of conceptual change show that if students hold misconceptions of conflicting models with new information, their ability to apply and transfer that information correctly will be impaired[9]. While much of the research has addressed situations where the conflicting models are incorrect[1], the existence of competing models interfering in problem solving domains is well documented[6].

This pilot study asks the question of whether prevalent mental models exist for data intensive software applications,

and if so, can those models be aligned with data structures introduced in early computer science classes.

3. RESEARCH DESIGN

Information was gathered from subjects through in-depth interviews. There were 8 subjects interviewed from a private research university(7) and a public university(1), varying in computer science experience and gender. The categories of computer science experience were:

- No Programming:the person had never taken a programming class.
- Programming/Non Major:the person had taken 1 or more computer science classes but was not a computer science major.
- CS Major:The person was an upperclassman pursuing a computer science degree.
- Faculty: the person was a faculty member in either CS or Engineering at the university and had an extensive programming background.

Subjects answered questions about the data intensive programs that they used on a regular basis. Prior to being interviewed participants completed an intake survey that asked how often they used programs such as email, iTunes, mapping software (like Google Maps), a cell phone address book, instant messenger, Facebook, and You Tube.

Participants were asked the same four questions about each application they used frequently (more than once a week). The questions were designed to find out about both the structure and relevant operations on the data held within the mental models. The questions were:

1. How do you normally interact with the program?
2. How do you keep your information organized? If you had to imagine how the computer saves or stores your information what would that look like?
3. If you want to get to one particular piece of information, how do you get there?
4. Imagine I want to change the way you interact with the software program so that you can see only one piece of data at a time. Which of the following four buttons do you think would be the most used? Get First, Get Last, Get Item Number , or Search? Which of the buttons do you think would be the easiest to make?

The questions were always asked in this order for all subjects and applications. As participants answered the questions, the interviewer may have asked for clarification if the answer was vague.

4. CODING

The data was coded at the utterance level from each subject. For the purposes of this analysis an utterance was defined as an idea that was expressed either in a statement or series of statements. A total of 228 utterances were recorded and coded. For each utterance, the application they were discussing, the question they were answering, and the time on the video was recorded. Each statement was coded as a reference to one or more data types based upon a table of

structures and important behaviors of common data types. Data types included in the coding were Array, List, Map, Set, Matrix, Stack/Queue, and Graph.

For example the following student statement was coded as a Set because it referred to the unordered nature of the data in her perception. "Its like a big amorphous cloud and you as the user get to choose the way that you organize it." The utterance "I think its an actual list - most recently added, and from that list other lists are referenceable, like categories" was coded as both a list and also stack/queue. It made a reference to the word list (which implies a linear structure), as well as most recently added, which implies a time sequence.

The data was also coded based upon the interactions you could have with the information. A table of important behaviors for each data structure was created and used to code the utterances which dealt with data interactions. The java api was used as a reference for the kind of operations attached to each data structure. The concept of search, for example, was coded as both a List and a Set operation because both of those data structures contain a built in search feature. For example, the student utterance "Normally IŠm looking for a specific friend, I donŠt browse around too often, I just use the search bar" was coded as List and Set for this reason.

The table of structure vs. behavior is available at the author's web site in the appendix of this paper at

<http://www.cs.cmu.edu/~lsudol>.

5. RESULTS

Although the small subject size makes it hard to generalize, the consistency across very different subjects in a variety of domains offers strong suggestions of underlying structures.

5.1 Application Data

The 8 subjects were asked about applications they used more than once a week. There were 8 different applications that were covered, with an average of 4.6 applications per person. The four questions were asked about each application that participants indicated they used frequently on an intake survey. For each application the number of relevant utterances range from 2 to 12 per person.

5.2 Data Type Difference

There were several patterns that emerged across people and applications. The common language for List and Set included reference to the operation of searching, the ability to see if an item was contained within the data, or to look at all the items at once. Statements coded as stack/queue frequently involved the mention of a "first" or "last" item and the idea that order was preserved was very strong. Statements coded for arrays indicated rigidity (in terms of size or ordering) and indexes for each item.

5.3 Overall Utterance Count

As you can see in Figure 1; List and Set had the most utterances of all the data structures. The graph shows by data type the count of the utterance across all subjects and applications. Remember that one utterance may be coded in different categories.

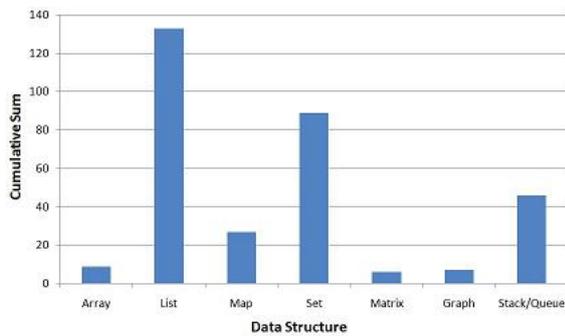


Figure 1: Utterance count by data type

5.4 Subject Background

The results can also be looked at by subject background. Presented here are the percentages since there were different numbers of utterances total per subject.

You will notice in the data from Table 1, that although arrays are often the first way that students are taught to store multiple records within a program, there is little evidence from this pilot that people have prevalent models for arrays. Even after taking a computer science course, people still tend to refer to abstract, dynamic models of data.

Table 1: Table 1: Data Structures by Subject Background

	No Prog.	Non Major	Major	Faculty
Array	3%	5%	2%	0%
List	38%	28%	34%	45%
Map	6%	5%	7%	17%
Set	33%	17%	26%	20%
Matrix	0%	3%	2%	2%
Graph	0%	3%	5%	1%
Stack/Queue	10%	20%	12%	9%

Although these results should not be considered generalizable, faculty members showed a strong bias towards List based data structures, while students who had never taken computer science were much higher for set or unordered structures. While these two results would be difficult to generalize to a larger population, it does motivate further study into what types of models people hold of data, and how that can be mapped to the type of data storage that students could encounter in an early programming course.

5.5 Gender Results

Table 2 shows the results by gender. While the numbers vary slightly the same overall pattern of increase in the List and Set categories holds across both genders. While we often encounter discussion the gender differences between students studying computer science, there seems to be little evidence at this time for the idea that men and women perceive the information in the computer differently. However, in future studies gender will be recorded to validate this on a larger, more diverse population.

5.6 Results by Application

One question that needs to be explored is whether it was the applications discussed that prompted the pattern observed, or whether common models exist across all applica-

Table 2: Table 2: Results by Gender

	Male	Female
Array	2%	4%
List	30%	36%
Map	9%	3%
Set	24%	27%
Matrix	2%	1%
Graph	2%	2%
Stack/Queue	15%	14%

tions that get used on a regular basis. Table 3 shows the breakdown by utterance count across all of the applications.

Table 3: Table 3: Utterances by Application

	Array	List	Map	Set	Matrix	Graph	S/Q
Torrent	1	12	2	4	0	0	3
CellPhone	1	21	6	12	0	0	7
Email	3	34	4	29	1	0	19
Facebook	1	23	6	15	1	2	9
Google Maps	2	15	1	8	4	4	2
IM	1	4	4	5	0	0	3
iTunes	0	17	3	14	0	0	2
YouTube	0	7	1	2	0	1	3
Totals	9	133	27	89	6	7	48

Despite both the differences in the visual and interactive aspects of the data contained throughout the different application, the pattern of responses remain the same. A majority of the utterances code as either List or Set based data structures. This consistency, despite the different software applications that were discussed implies a commonality to the models that the students hold.

6. OPEN QUESTIONS

This pilot study opens several questions for discussion.

- Does taking a computer science course (or many courses) change the way which we perceive data storage and interactions?
- Are there mental models that are prevalent in the student population before they take a computer science course?
- Should introductory education focus on utilizing pre-existing models to sequence our instruction, or should we teach our standard curriculum because it is the standard curriculum?
- What other features of the mental models should be explored in future studies?
- If these results are replicated in a larger study, what implications does that have for computer science education and computer science education research?

7. ACKNOWLEDGMENTS

The research reported here was supported by the Institute of Education Sciences, US Department of Education, through Grant R305B040063 to Carnegie Mellon University.

The opinions expressed are those of the authors and do not represent the views of the Institute or the US Department of Education

8. REFERENCES

- [1] M. T. H. Chi, J. D. Slotta, and N de Leeuw. From things to processes: A theory of conceptual change for learning science concepts. *Learning and Instruction*, 4:27–43, 1994.
- [2] National Research Council. *How People Learn: Brain, Mind, Experience and School*. National Academy Press, 2000.
- [3] Kenneth Craik. The comprehension of the everyday physical environment. *Journal of the American Planning Association*, 34:29–37, 1939.
- [4] Linxiao Ma, John Ferguson, Marc Roper, and Murray Wood. Investigating the viability of mental models held by novice programmers. *SIGCSE 2007: Proceedings of the 38th SIGCSE technical symposium on computer science education*, pages 499–503, 2007.
- [5] Richard E. Mayer. The psychology of how novices learn computer programming. *ACM Comput. Surv.*, 13(1):121–141, 1981.
- [6] George J. Posner, Kenneth Strike, Peter Hweson, and William Gertzog. Accommodation of a scientific conception: Toward a theory of conceptual change. *Science Education*, 66:211–227, 1982.
- [7] Beth Simon, Dennis Bouvier, Tzu-Yi Chen, Gary Lewandowski, Robert McCartney, and Kate Sanders. Common sense computing(episode 4): Debugging. *Computer Science Education*, 18:117–133, 2008.
- [8] Beth Simon, Tzu-Yi Chen, Gary Lewandowski, Robert McCartney, and Kate Sanders. Commonsense computing: what students know before we teach (episode 1: sorting). In *ICER '06: Proceedings of the second international workshop on Computing education research*, pages 29–40, New York, NY, USA, 2006. ACM.
- [9] S. Vosniadou. The cognitive-situative divide and the problem of conceptual change. *Educational Psychologist*, 42:55–66, 2007.
- [10] Stella Vosniadou. Mental models in conceptual development. *Model Based Reasoning: Science, Technology, Values*, 1, 2002.
- [11] L. S. Vygotsky. *Mind in Society: The development of higher psychological processes*. Harvard University Press, 1978.
- [12] R. M. Young. *Mental Models*, chapter Surrogates and Mappings: Two Kinds of Conceptual Models for Interactive Devices, pages 32–52. Erlbaum, 1983.