Declarative Programming and Cognitive Load: Strategies for Reducing Task Complexity

Leo Vijayasarathy and Gretchen Irwin
Computer Information Systems
College of Business
Colorado State University
For Collins, CO 80523, USA

Database management remains a core component of Information Systems (IS) curriculum. Students majoring in IS are typically required to complete one or more courses that cover database design, development, use and administration. While the content and learning objectives in these courses is expanding to include models, techniques and technologies to manage unstructured data, there is still a strong emphasis on the relational model and structured query language (SQL). SQL is a standardized programming language for defining, manipulating, querying and administering relational databases. In contrast to imperative languages, that provide step by step instructions to accomplish a particular task, SQL can be used as a declarative language (especially for querying), to specify the task to be done without necessarily spelling out the details of how to do it (Chamberlin, 2012). Although SQL can be used for querying without prior knowledge of programming constructs such as variables and control structures, learning to write correct queries for complex tasks is a challenge.

There are two primary sources of complexity in query tasks – transformational/lexical and structural. Transformational or lexical complexity is associated with the chasm between the query request and the target data model, and is a function of the effort required to map a query task to relevant elements in the data model. Structural complexity is attributable to the gap between the query request and the target SQL statement, and is related to the effort needed to represent the query task in terms of the required SQL clauses, functions, and keywords.

Cognitive load theory (CLT), which elaborates on human cognitive processes and the interactions between working and long-term memories, is used to research and design instructional materials and strategies to facilitate complex learning (Sweller, 2005; Sweller and Chandler, 1994; van Merriënboer and Sweller, 2005). According to CLT, humans rely on long term memory and its repository of knowledge stores or schemata to work in conjunction with working memory, which is limited in its capacity and duration for dealing with novel data, to

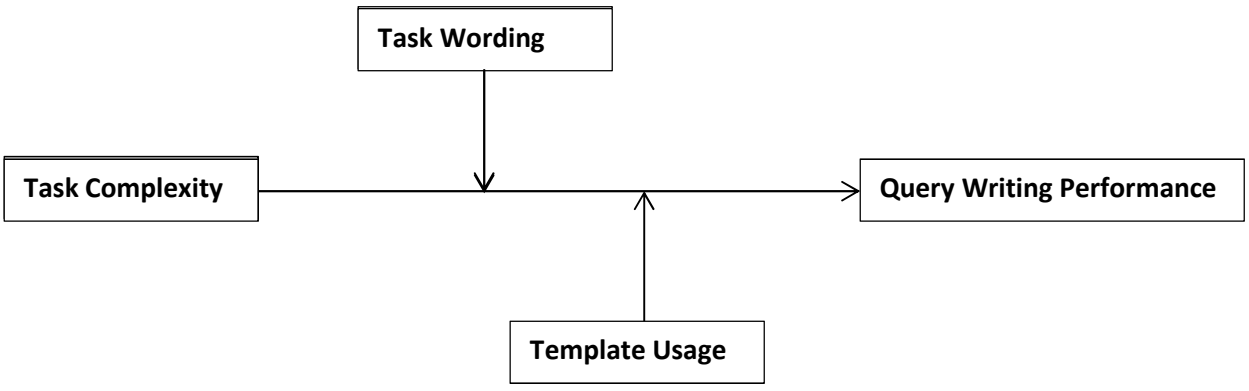
recognize, process and solve problems. Since CLT posits that expertise is a function of schemata complexity and automation, instructional design and strategies that promote schemata development in long term memory while easing the demands on working memory can be beneficial to learning complex tasks. The cognitive load on working memory can be intrinsic (function of the interactivity among the elements to be learned), extrinsic (tied to how and what learning materials are presented; irrelevant and inappropriate materials increase extrinsic load), and germane (associated with activities that encourage schemata construction).

Our research agenda is to develop instructional methods that are effective in teaching declarative programming. We are conducting studies to test the efficacy of different strategies in reducing cognitive load for novice SQL learners, and one of our experiments addresses the following two research questions:

1. Do novices perform better when query tasks are worded as pseudo-SQL?
2. Do novices perform better when asked to complete a template with guiding questions prior to writing the query statement?

As shown in in Figure 1, we expect task wording and template usage to moderate the relationship between task complexity and query writing performance by mitigating the transformational/lexical and structural sources of complexity respectively.

Figure 1



Our study participants were Computer Information Systems and Computer Science students enrolled in two sections of a junior-level database management course at a large US public university. We used a 2*2*2 repeated measures factorial design for our study. Each of the three

factors has two levels - high vs. low task complexity, managerial-English vs. pseudo-SQL task wording and template use vs. no template use. A balanced Latin-square design was used to minimize carry-over effects (Pezzullo, 2008; Williams, 1949). We developed a custom application that a) provided an interface for subjects to perform the experimental tasks, and b) had back-end code to capture and record relevant performance data for each subject including the query tries, times taken and errors generated.

We are analyzing the data gathered from this experiment, and will present the results at the conference.

References

1. Chamberlin, D.C., 2012. Early history of SQL. *IEEE Annals of the History of Computing* 34 (4), 78-82.
2. Pezzullo, J.C., 2008. Latin squares for constructing "Williams Designs", balanced for first-order carry-over (residual) effects. URL: http://statpages.org/latinsq.html.
3. Sweller, J., 2005. Implications of cognitive load theory for multimedia learning. In R.E.Mayer (Ed.), *The Cambridge Handbook of Multimedia Learning*, Cambridge University Press, New York, NY, 19-30.
4. Sweller, J. and Chandler, P., 1994. Why some material is difficult to learn. *Cognition and Instruction* 12 (3), 185-233.
5. Van Merriënboer, J.J.G. and Sweller, J., 2005. Cognitive load theory and complex learning: Recent developments and future directions. Educational Psychology Review 17 (2), 147-177.
6. Williams, E.J., 1949. Experimental designs balanced for the estimation of residual effects of treatments. *Australian Journal of Scientific Research* 2 (3), 149-168.