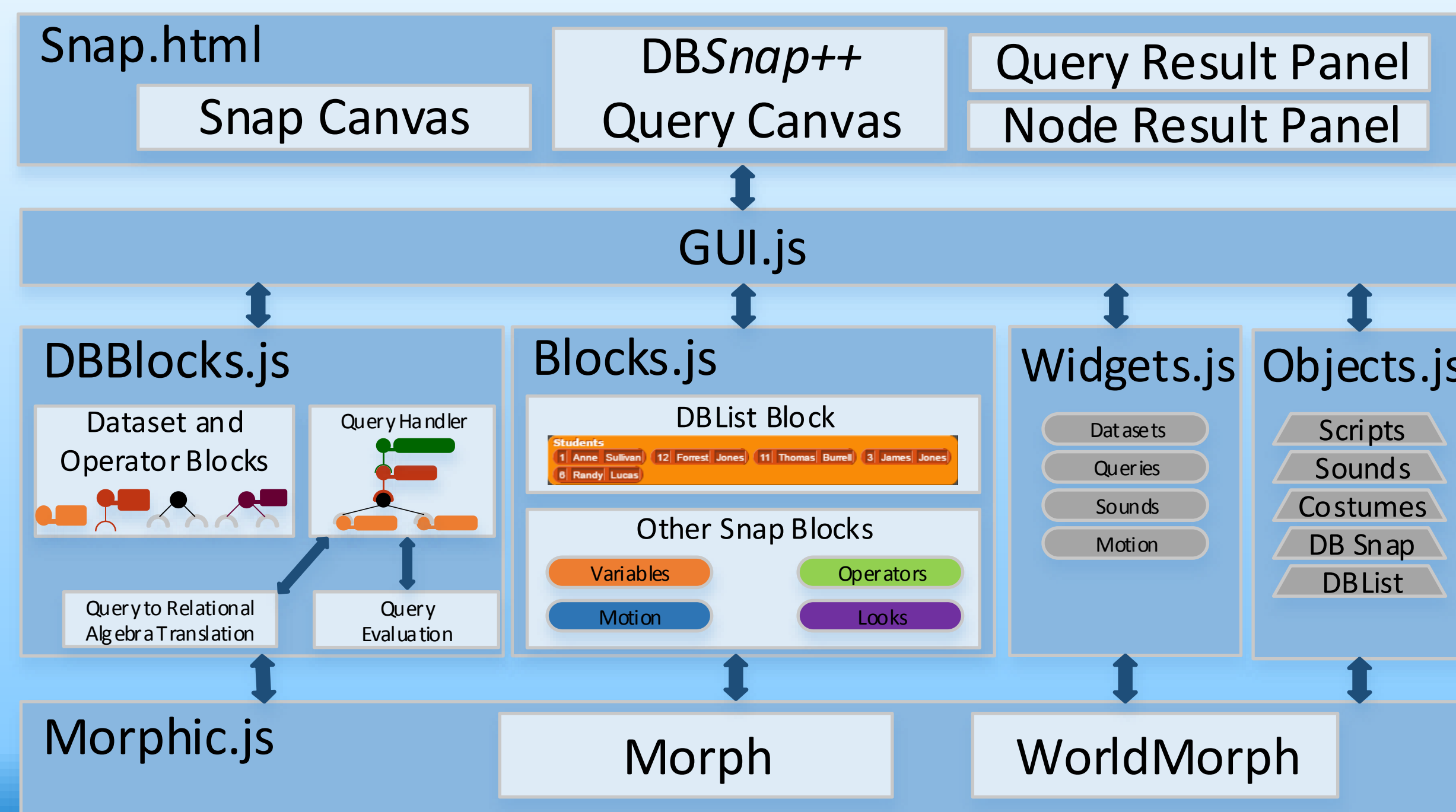


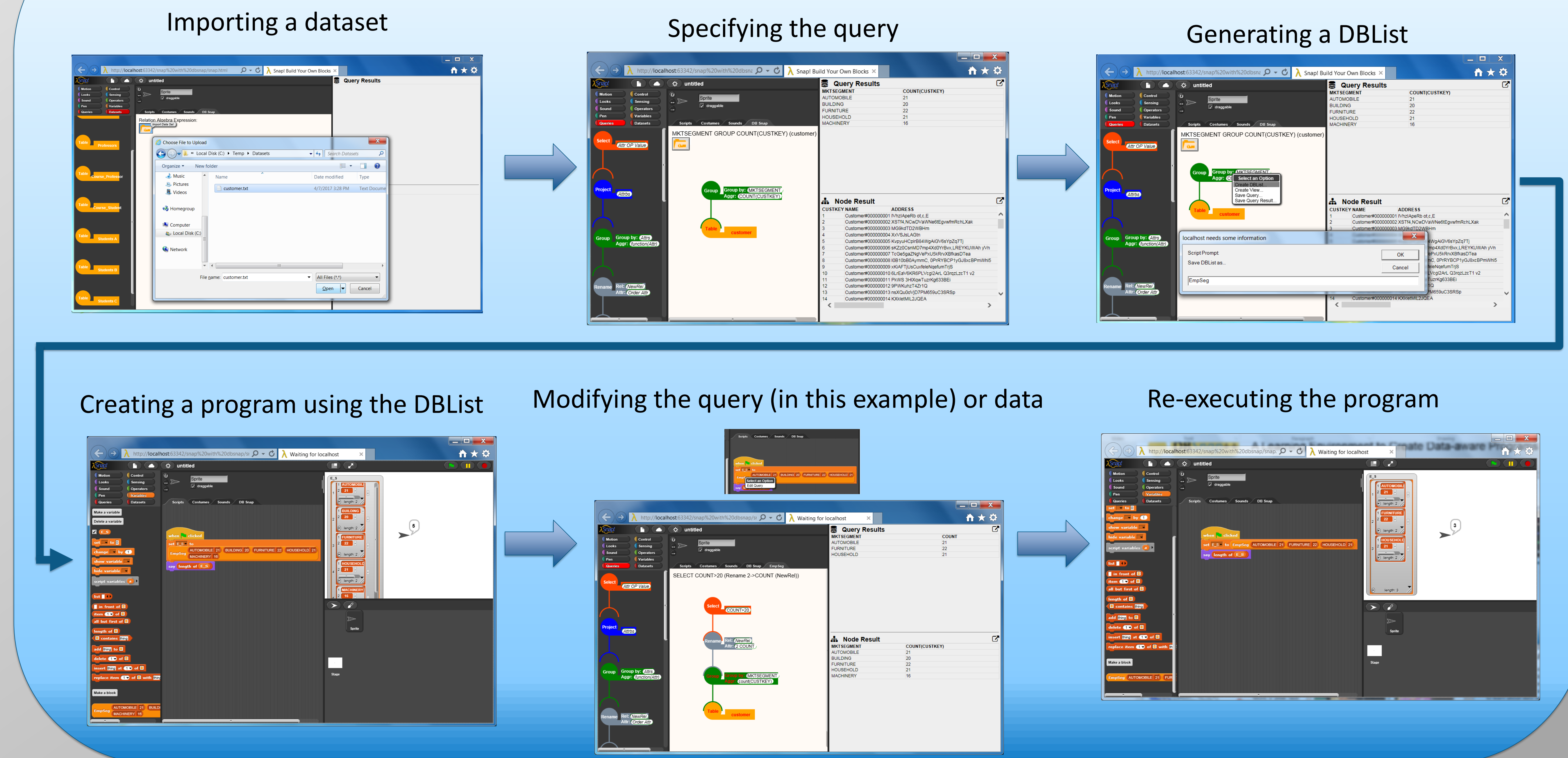
Overview

- Computer Science Education research has focused on studying block-based programming environments where programs are created by connecting blocks.
- Most of these environments support conventional (imperative) programming instructions.
- While some systems have been proposed to enable the specification of database queries, there is significantly less work on the development of integrated systems to build full data-aware programs.
- This poster introduces DBSnap++, a web-based environment that enables the specification of programs that integrate conventional programming instructions and database queries.
- DBSnap++ supports the specification of intuitive database query trees and a new type of list that dynamically gets its content executing a database query.
- DBSnap++ supports the manipulation of database tables as well as the creation of programs that generate different results when the underlying data changes.
- DBSnap++ aims to have a transformational effect on data-aware programming learning.

Architecture



Creating a Data-aware Program



Supported Query Blocks

Select: $\sigma_{\theta}(R)$. This operator selects all the records of relation R that satisfy the predicate θ .

Cross Product: $R \times S$. This binary operator pairs each record of R with each record of S.

Projection: $\pi_{a_1, \dots, a_n}(R)$. This operator removes all the attributes of R not contained in a_1, \dots, a_n .

Theta-join (θ -join): $R \bowtie_{\theta} S$. Returns a similar result than the Cross Product but selecting only the rows that satisfy the predicate θ .

Rename: $\rho_{(i_1 \rightarrow b_1, \dots, i_k \rightarrow b_k)}(R)$. This operator changes the name of relation R to S and the name of the attribute at position i_j to b_j .

Natural Join: $R \bowtie S$. This operator is similar to the θ -join where the θ predicate is the equality of all the common attributes between R and S.

Aggregation: $g_1, \dots, g_m, G_{f_1(a_1, \dots, f_k(a_k))}(R)$. This operator groups the records of R forming a group for each unique occurring permutation of the grouping attributes g_1, \dots, g_m . If aggregated, computes each group under those circumstances.

Set operators: DBSnap also supports common set operations such as Set Union (RUS), Set Difference (R - S) and Set Intersection (R \cap S).

Sample Programs

Generating Bar Graphs

A program that displays data using a bar graph

Database: University Dataset
Data tables: Students
Program: Shows the number of students in each grade level. A grade level is represented by a sequence of line segments. Each line segment represents a student in that level.

The Sound of Data

A program that plays different notes based on retrieved data

Database: University Dataset
Data tables: Course_Student
Program: Represents the number of grades received by the students of a given class. Plays a beep for each student and different notes for different letter grades.

Future Work

- Complete final implementation tasks
- Make DBSnap++ publicly available
- Evaluate the effectiveness of DBSnap++ as a learning tool
- Compare DBSnap++ with alternative solutions and ways to learn about data-aware programming
- Write a research paper with the results of our work

References

- Y. N. Silva, J. Chon. Querying Databases by Snapping Blocks. The 31st IEEE International Conference on Data Engineering (ICDE), Seoul, Korea, 2015
- Y. N. Silva, J. Chon. DBSnap: Learning Database Queries by Snapping Blocks. The 46th ACM Technical Symposium on Computer Science Education (SIGCSE), Kansas City, Missouri, USA, 2015
- J. Gorman, S. Gsell, and C. Mayfield. Learning relational algebra by snapping blocks. In ACM SIGCSE, 2014.
- J. D. Hellmann. DataSnap: Enabling Domain Experts and Introductory Programmers to Process Big Data in a Block-Based Programming Language (Master's thesis). Virginia Tech, Virginia, USA, 2015.
- E. Modrow. SQLsnap!. <http://snapextensions.uni-goettingen.de/>.
- J. H. Maloney, K. Peppler, Y. Kafai, M. Resnick, and N. Rusk. Programming by choice: Urban youth learning programming with scratch. In ACM SIGCSE, 2008.
- C. North and B. Shneiderman. Snap-together visualization: Can users construct and operate coordinated visualizations? Int. J. Hum.-Comput. Stud., 53(5):715-739, 2000.