

XQuery to SQL Translation module for Shrex

Fall '08 CS6530 Project, Pramod Sanaga

Progress report on October 19th:

- Completed reading the W3C Schools tutorial on XPath, XML and XML Schema.
- Completed reading the W3C Schools tutorial on XQuery.
- Completed reading the related work papers listed on the course website and referenced in the project proposal.
- Created an Eclipse project for the Shrex source code, compiled it without any errors. Got a high level view of the working of the code base and that of the XPath query translation in particular.
- I noticed that Shrex currently supports XPath queries without using a generic parser to handle generic queries – it uses a single method in “XPathQuery.java” to translate just two kinds of queries into SQL – simple path expressions and expressions involving wild cards.
- I came to the conclusion that this approach is not going to be enough to support a meaningful subset of XQuery (including the FLWOR construct). The correct way to deal with the XQuery -> SQL translation is to make use of an XQuery reference parser.
- The XQuery parser should return an intermediate representation (IR) of the XQuery expression, which can then be used along with Shrex annotations to generate a series of join expressions on the involved tables.

Work going forward:

- Finalize the choice of an XQuery parser and figure out how to use the normalized form of its IR to generate an SQL mapping.
- Finalize the subset of XQuery that I would try to tackle. Juliana suggested that I choose a smaller subset than the one proposed in the initial project report. I am aiming for the FLWOR construct at the least, and to add the ‘for’ and ‘If-else’ constructs if feasible.
- Naively translated XQuery -> SQL expressions can result in a large number of joins, which could in theory be optimized (according to the related work publications). The current plan is to not attempt this part – though this might change in the future.