Introducing Database Slicing Method and Context-Free Export/Import

Dávid Tengeri

Nowadays more and more software packages use a database of some kind. This database could be very large, hence it might be very difficult to backup, to migrate or to debug. In such cases it would be useful to find the relevant slice of a database, and backup, migrate or debug only that part.

One way to reduce the size of the backup is to identify the really meaningful part of the data records. This can be called database slicing, which is similar to program slicing ([1], [3]). We used the elements of program slicing to describe the links between the tables of a database. It is easy to select those records from the database that are really needed for making a backup of important data, migrating data from one database to another or finding data-related bugs in the database with our new approach. There are two types of slicing, namely static and dynamic used in program slicing. We redefined these methods in the context of databases. Static slicing works with tables of the database, while dynamic slicing is based on the records of tables. We devised slicing algorithms for these types as well. The slicing algorithms use a dependence graph as their input and produce a slice of database. The algorithms are based on breadth-first search method.

There are a lot of papers which describe how to save your database content, but these methods saving the whole content of the database or they don't define how can we reload the exported data ([2],[4]). The process of context-free export/import enables moving the results of database slicing from one database to another without heavily depending on the IDs of tables. With this method we can select a small piece of the database content and migrate it somewhere else, without losing connections between the records in the slice and duplicating data in the target database.

Owing this, we defined an XML-based language for storing the result of slicing algorithms. With our document we can describe the schema of a database separate from the database system that we are using and we store only the columns of the tables which are necessary to import the data correctly. The schema of the database determines the structure of the exported data.

Later, we implemented the slicing method and the exporting/importing process using the Drupal content management system.

References

[1] David Willmor, Suzanne M. Embury, Jianhua Shao: Program Slicing in the Presence of Database State, 2004, *20th IEEE International Conference on Software Maintenance*

[2] Dongwon Lee, Murali Mani, Frank Chiu, Wesley W. Chu: NeT & CoT: Translating Relational Schemas to XML Schemas using Semantic Constraints, 2002

[3] Frank Tip:A Survey of Program Slicing Techniques, 1995, *Journal of programming languages, Citeseer*

[4] Joseph Fong, San Kuen Cheung:Translating relational schema into XML schema definition with data semantic preservation and XSD graph, 2004, *Information and software technology, 2005, Elsevier*