Data Documentation and Retrieval Using Unity in a UniVerse® Environment

Progress report – Spring 2002

May 15, 2002

Jose JIMENEZ

University of Iowa, Iowa City, Iowa

jose_jimenez73@hotmail.com

Goals

The goal of the first phase of this project as defined in earlier documentation[1] was to create preliminary connectivity to the UniVerse® environment from Unity. Initially this was to be achieved by mapping accessible data fields using Unity. An additional goal was to document some of the inherent difficulties in accessing multi-valued information via ODBC.

Details

Environment

In order to achieve the goals of phase one, a test environment was set up with a limited number of tables to access. The tables chosen were product and customer information files. In order to test accessibility via ODBC, Microsoft Query was used in conjunction with Microsoft Excel to access data from the test database. This gave a baseline for whether the UniVerse® database was visible to a standard ODBC based query. Using these tools some immediate problems came to light with the way some dictionary items were set up. An error was encountered when dictionaries in the UniVerse® test environment were set up as right justified and were not numeric fields. The ODBC client for UniVerse® seemed to be reading the right justification as meaning the field was numeric. This caused a type mismatch when reading mixed data in the field. This will be important for the next phase of the project which will be discussed later. To complete the setup of the test environment, some of the fields available to the ODBC client were modified to alleviate the field data type problems.

Unity

Once the test environment was setup, connectivity was tested using Unity and failed. In troubleshooting the problem with connectivity, it was found that Unity did connect to the data source, but failed to read the table and field information via the ODBC connection. One of the first issues encountered when examining the source of the

problem more closely was the close ties between Unity and Microsoft Access databases. In development and testing of Unity, Access databases were used most often and because of this, some of the source code for Unity was written specifically for Access. This caused Unity to mistakenly report that it was looking for a file with "mdb" as its extension which is the file extension for Microsoft Access databases. This was a minor issue and was remedied quickly.

A more pervasive issue was Unity's use of ODBC level 3 calls to the data source. ODBC level 3 is the current release level for ODBC drivers. The ODBC driver in use with the UniVerse® test environment was only level 2 compliant. This caused some of the data layout discovery features of Unity to fail. Unfortunately, this meant that one of the basic building blocks of Unity data access, the source definition file, failed to be completed. In order to correct the problem in the data source discovery, Unity was modified to remove the incompatible calls.

Once the data source file was built, work began to ensure the rest of Unity worked with the UniVerse® database. In order to do this, a specification was built up from the source file and a schema was defined from the specification file. Initially, only one data file was defined with any amount of detail, in order to continue testing. The next connectivity test was execute a query using Unity against the Universe database. Again it was discovered that Unity was using ODBC level 3 calls to do the query execution. Unity was not using the information discovered during the source discovery when executing the query and was unnecessarily performing field information retrieval before accessing the data from the tables.. In order to get beyond this, the ODBC level 3 calls were replaced with similar calls conforming to the ODBC level 2 standard.

Once Unity was able to execute queries against the UniVerse® database, the next step was to attempt a query that included a join across tables. This did not work initially but was not a problem with Unity. Instead, the data specification for the joins had not been done correctly by the user.

Further work for Unity

Unity should be expanded to make it compatible with more data sources. Similar to the problem found when trying to work with the UniVerse® database, other types of databases may not be compatible with ODBC level 3 or level 2. Unity could be modified by creating inheritance classes which could be called for different levels of access. This would entail creating a master class that would have the calls for ODBC level 1 and subclasses that would override certain functions such as the data field discovery. Another use for these would be the addition of ODBC driver specific calls where necessary. For example, UniVerse® has two different methods of accessing tables. These two methods cause multi-valued fields to be returned differently. If one is preferable to another, opening the data source should be done in such a way to take advantage of the correct method.. This would require different calls that are not necessary for other database types and would best be separated from the rest of the code by creating inheritance classes.

Further work for the project

The next phase for the project is to increase the amount of information available to Unity from target database. More tables and fields will be added to the @select record of each table in order to increase the amount of data available. In order to do this without

doing it manually for each field, it will be necessary to create a program to choose among the different dictionary items for a data field. This program will run in the UniVerse® environment and populate the @select item as necessary. This phase of the project will begin during the summer of 2002.

The third phase of the project may also be started during the summer. This phase is to create a source, specification and schema for the database once more. This would entail a repeat of some of the work done during spring 2002, but would not have to overcome the same problems. This would give a baseline for the amount of documentation possible manually using Unity.

The fourth phase will be attempted during the fall of 2002 and will entail increasing the amount of automated documentation available. The problems encountered this spring in using Unity with UniVerse® used a lot of the time allotted to phase one and two of the project. It is possible the fourth phase, which deals mainly with semi-automated documentation of databases using Unity and yet to be created support programs, may not be completed by the fall and may have to be left for further work.

Summary

Now that ODBC connectivity has been shown to work with Universe, the next phase of the project can begin. The next phase of the project will focus on increasing the reliability and amount of data available to Unity via the ODBC driver. The recently completed phase of the project suggested improvements for Unity as well as possible problems and solutions for the rest of the project.

References

[1] J. Jimenez: Data Documentation and Retrieval Using Unity in a UniVerse® Environment. February 21, 2002.