ConventionOverConfiguration

ConventionOverConfiguration

This concept reflects the notion that configuration (defining a bunch of stuff in a side file) can be avoided by following convention (adhering to prescribed patterns) and that this is generally a good thing. The RDB DAS has several reasonable conventions in place and using these can significantly reduce a developer's work-load.

As an example, the RDB DAS provides a straight-forward, one-to-one mapping of database tables to DataObject Types and from table columns to DataObject properties. If a user provides no mapping information to the contrary, a graph of DataObjects returned by the DAS will directly map Types /Properties to the Tables/Columns queried.

For example, consider a database with the following table:

```
TABLE CUSTOMER
     ID         INTEGER
     LASTNAME   VARCHAR(20)
     ADDRESS    VARCHAR(30)
```

If an application uses the DAS to query this table with the following SQL statement:

```
SELECT * FROM CUSTOMER
```

The DAS will return a graph of DataObjects each of Type CUSTOMER and each instance in will have three properties: (ID, LASTNAME, ADDRESS). Here is the example code. Note that no configuration data is provided at all.

```
DAS das = DAS.FACTORY.createDAS(getConnection());
Command readCustomers = das.createCommand("select * from CUSTOMER");
DataObject root = readCustomers.executeQuery();
```

Another piece of convention followed by the RDB DAS has to do with database table keys. When asked to "apply changes", the RDB DAS scans the SDO ChangeSummery and generates the set of INSERT/UPDATE and DELETE statements needed to flush the graph changes to the database. To generate the correct statements the RDB DAS needs to know how a DataObject's properties map to a database table key. In typical Object/Relational mappings frameworks, this information is provided in a config file with something like this:

```
<Table tableName="COMPANY">
      <Column columnName="ID" primaryKey="true"/>
      .
      .
      .
   </Table>
```

However, the RDB DAS understands is that, in the absence of user provided cofiguration data, a DataObject property named "ID" maps to a key column named "ID" in the database. If a developer follows this convention (naming key columns "ID") then this key-mapping information need not be specified in a configuration file.

The RDB DAS also has convention in place for mapping DataObject relationships. If a database table (yyy) has a column named xxx_ID then the DAS will assume, in the absence of config data to the contrary, that there is a one-to-many relationship from table xxx to table yyy and that the key column for table xxx is named "ID".

There is more "convention over configuration" coming. We are currently looking into support for optimistic concurrency control conventions and other areas.