

# Study on the possibility of automatic conversion between BusinessObjects Universes and Oracle Business Intelligence Repositories

Olti QIRICI

Department of Informatics

Faculty of Natural Sciences, University of Tirana

Tirana, Albania

olti.qirici@fshn.edu.al

## Abstract

The decision on which Business Intelligence solution should be selected is very difficult. Sometimes it is required to switch between these solutions. Here we will try to check the features that some commercial products provide us, that help automatize the process of models creation. We will check for possible mappings between these solutions toward a final automatized conversion process which would simplify the process of products selection without enhanced efforts on models conversion. Both ways of conversion have been investigated and are feasible but currently it seems that one of the automation ways is possible with least effort.

*Keywords — BI, BO, BusinessObjects, OBIEE, universe, repository, automatic*

## 1 Introduction

The process of selecting which Business Intelligence tool will be used is not just a matter of which technical solutions fits most to a current situation. It is true mainly because all the possible BI solutions, try to provide all those tools and functionalities which are necessary for these systems. Sometimes these tools might provide some extra features which make it simple to perform some operation. According to Turban et al [Tur11] there are four main features which are prerequisites for a system to be considered as a BI solution which are:

- I. Data Warehouse
- II. Business Analytics
  - a. Reports and queries
  - b. Data, Text and Web mining and other sophisticated mathematical and statistical tools
- III. Business Performance Management

## IV. User Interface

Vast majority of business BI solution today has all of them implemented. At least the products which we are discussing here, have them all implemented.

So which might probably be the most important reason for choosing a specific product could be the price of a specific product or how strategically the choice of a given product might be combined with specific discounts on sales prices and favor a choice over another. Generally BI products are quite expensive which might make even small discounts considerable.

Sometimes, enterprises prefer to choose one solution over the other because during the process of infrastructure transformation, it is required for the sake of uniformity to switch between different platforms. If a company or state enterprise decides to change the technological providers, business intelligence tools, which represent these transformation from the end user perspective on reporting and analysis, would not escape these major changes, even for the reason of uniformity.

But BI models require a specific amount of time to be developed. As each information technology project, this time is divided in requirement specification gathering, design of the solution, development, testing and maintenance. When the management changes strategy and product, of course a migration of the BI model is needed. But not all these steps have to be performed as a whole, while the longest step during the migration might be the development phase. In order to shorten this phase, it is required to produce an automation tool which would perform this transfer with little supervision required. This automation could be both ways or single way extracting information from a product and recomposing it through feeding the other product. By so doing, it would be simpler to facilitate the decision on business intelligence solution migration without major changes.

Even partial migration would be welcomed since the manual recreation of such solutions is generally a long task with heavy handworks. Saving of time to the IT department would enhance their acceptability of the change and would replace the efforts on the other phases of the migration project.

First lets have a look on each of the products we are referring to.

## **2 BusinessObjects compared to Oracle Business Intelligence**

Business Objects is the major product developed by SAP which provides it's users with the most advanced Business Intelligence features. It fulfills with specific tools all the requirements for developing the business intelligence basics. It is possible to build queries (free hand SQL queries or through the metadata creation), this can be fed through a data warehouse, additional data, text and web mining techniques can be implemented for a more accurate analysis of an unstructured information and it has a SAP BPM tool. Of course it is enriched with a graphical user interface which is depending on the accessibility needed for each of the tools. It has desktop developed tools, basically tools required to perform heavy, not-collaborative tasks which require a single user to utilize it at a time. It also provides web components or tools for the tasks which are used massively by the users and which are basically used for analysis and report generation. Currently, this SAP product has reached the version of BO 4.2.

Oracle Business Intelligence from the other side is the major Business Intelligence tool developed by Oracle. The current version is 11g.

“Oracle Business Intelligence Enterprise Edition 11g (OBIEE) is an unmatched and comprehensive business intelligence and analytics platform that delivers a full range of capabilities including interactive dashboards, ad hoc queries, mobile analytics, notifications and alerts, enterprise and financial reporting, scorecard and strategy management, business process invocation, unstructured search and collaboration, integrated systems management and more.” [Ora13]

This shows basically main features offered by OBIEE which are comparable to those offered by BO.

What we have to be focused here it is the metadata which are used by these products in order to feed the tools when data are not directly extracted through free hand SQL. These metadata organizations are called Universes and Repositories respectively for the BO and the OBIEE. This intermediate layer between the different data sources and the processed information for an organization is a specific complex multitier metadata which stores information on these data

sources, how these sources are interlinked and how this information is mapped to business rules.

The BO Universes and the OBIEE Repositories are designed through some specific tools, part of the product which follow up the design process through these layers. These tools respectively are called as Universe Designer and Administrator Tool. Both these tools are desktop applications which are used generally by a single user, even though there exists collaboration techniques and configurations which might be used in the cases of distributed development environment. But this collaboration is not very easy and sometimes it requires manual merging of the changes done by concurrent developers.

Hereafter we will specify the way these applications organize the information in multiple layers and how all this information can be developed in these tools from the data sources to the presentation manifestation of these data.

These tools have been used for long times now and no information is given by the companies which might make developers, integrators and researchers think that these tools will be switched by other tools (possibly web applications) or these tools will become deprecated in the near future.

Let us analyze this multilayer designed environment.

## **3 Layers of the metadata design**

Both the Administration Tool for OBIEE and the Universes Designer used by BusinessObjects, are separate tools used to design the metadata repository and are not part of the process of generating the reports and analysis. Actually they produce the core feeder for the intermediate layer of a OBIEE system, which is the information holder on the data organization and relationship.

Following is an image showing the Administration Tool which as can be viewed as a tool peripheral to the system but as producer of the Universe, creates one of the main elements used by Oracle BI Server, the Server which extracts from the Data Sources, the requests from the clients.

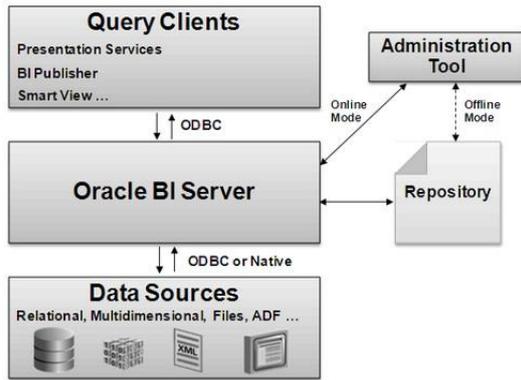


Figure 1. OBIEE architecture [Ger16]

In the following image it is shown that universes created from the Universes Designer are central also to the process of serving the client when referring to the BusinessObjects architecture.

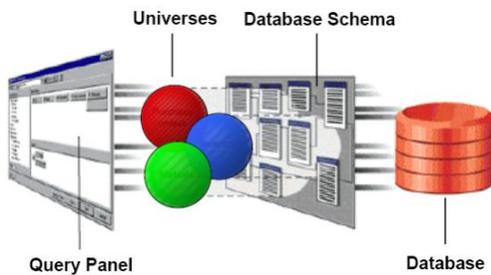


Figure 2. SAP Business Object Designer Architecture Diagram [Com]

But not just the Business Intelligence system is a multi-layer system itself. Also the tools provide the model through a three major layers.

- Physical Layer
- Business Model
- Presentation Layer

In order to have a working solution, the information should be passed through all these layers. All the three layers should be populated with the accurate information in order to provide a correct migrated metadata model between two different BI models.

These layers describe specifically the physical model, the business model and the mapping of information technical-wise and functional-wise, and the presentation layer which describes the model as it is viewed by the end user which is not required to know the structure of the sources and their dependences.

Hereafter we will be referring to the physical layer as the database layer since for this conference I have been focused mainly at databases as a source.

Both systems design their solution based on a group of elements which populate the various layers of their interface. There is a nearly one to one conversion of elements between OBIEE and BO. These elements will be matched hereafter following their specific layer belonging. So from the very basic (physical layer) to the presentation layer the most important elements might be (at least for a not so complex environment – since for complex tasks additional efforts might be executed manually on the environment to decrease the criticality of executing such transformations):

OBIEE element	BO element
BusinessModel	Classes
ConnectionPool	Connection
Database	Database
Dimension	Dimension
Measures	Measures
Details	N/A
LogicalComplexJoin	Joins
LogicalTable	Classes
LogicalTableSource	N/A
PhysicalDisplayFolder	Folder
PhysicalTable	Tables, Views, Derived Tables, Synonyms
PresentationCatalog	N/A
PresentationHierarchy	Hierarchy
PresentationTable	Classes
Schema	Schema

Table 1. Comparison between some of OBIEE and BO elements

As it can be seen by the above table, the number of elements specified by OBIEE is quite major from those of BO. But this doesn't make BO less suitable as a semantic layer interconnecting the Data Source with the user. For some of the elements there is seen no distinction between the presentation layer and the logical layer in BusinessObjects. In the literature, BusinessObjects is referred as “the powerful semantic layer which lets you create complex SQL statements, without you ever needing to know or write SQL” [How12]. Anyway from this standpoint, it can clearly be seen that even BO transfers the information through structures from the physical layer to the presentation one, since it creates independent information gathering from the data source information.

Following this logic, some information on different layers, clearly defined in OBIEE through different naming convention, in BO may be found under the same element. As was illustrated in the previous table, information on Classes (and sub-Classes unmentioned above) may be found in OBIEE as referring to logical and physical tables, as also can be linked to the Business Model itself.

Furthermore, OBIEE allows a wider range of organizing the information in subject areas, folders and subfolders for each of the layers, while BusinessObjects provides a more simplistic approach. As it can be seen in the above table, some of the terms we specified for OBIEE does not have any equivalent in BO.

Let us follow now with the analysis of the models for each of these systems.

#### 4 Oracle Business Intelligence Model

Following the previous discussion, here is shown a sample of built repository from a subset of tables regarding the PC Module from Oracle FCUBS 12.2. The schema regarding the relationship between these tables has been illustrated in this image.

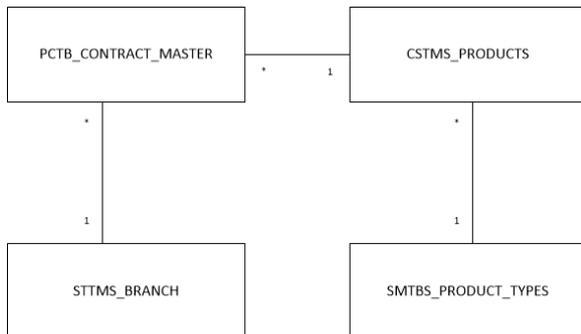


Figure 3. Relationship between tables in a subset of the PC Module

According to the business rules predefined by the need on querying information out of this partial database, a conceptual schema has to be designed on the result at the business layer.

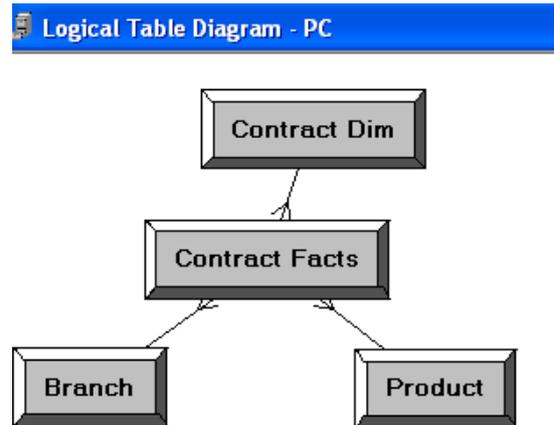


Figure 4. Conceptual relationship at the business end

It is necessary to develop in the Administration Tool a three layered structure which covers the information flow designing which will map the physical stored data to presented information, easily accessible by the user, straight through a business model which interlinks both ends.

In the following image we can see the result on such application for the above mentioned data model. This design will be used to explore the XML structure produced by the Oracle Administration Tool which describes this repository.

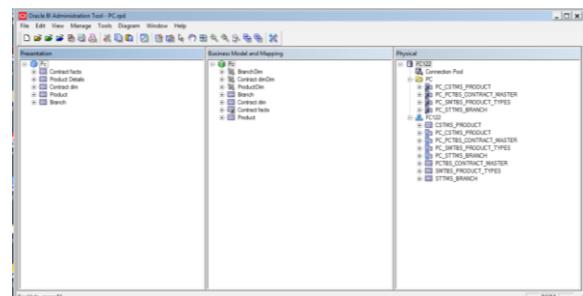


Figure 5. Three tier layer for the PC module in the Administration Tool

In the above image can be seen the three layers (from right to left: i) Physical Layer, ii) Business Model and Mapping and iii) Presentation Layer). This solution was developed following the sample from the Oracle manual [Ora12] suggests.

In order to produce an extract, which might be used as a universally integrated structure, without extracting the information from the repository

compiled file of Oracle, the only solution would be to export this compiled file to a readable format which might be created and exported, every time such task would be required. This option is a project of XML files, commonly known by Oracle as MDS XML Documents. These files are simple XML files, organized in folders which specify the different elements which should be defined to build a repository. These folders, which are then interlinked by IDs, uniquely identifying these elements and also which identify specific elements inside the XML files are alphabetically ordered as: BusinessModel, ConnectionPool, Database, Dimension, Group, InitBlock, LogicalComplexJoin, LogicalTable, LogicalTableSource, PhysicalDisplayFolder, PhysicalTable, PresentationCatalog, PresentationHierarchy, PresentationTable, Schema and Variable.

Name	Date modified	Type
BusinessModel	8/5/2016 2:49 PM	File folder
ConnectionPool	8/5/2016 4:39 PM	File folder
Database	8/5/2016 4:43 PM	File folder
Dimension	8/5/2016 2:49 PM	File folder
Group	8/5/2016 2:49 PM	File folder
InitBlock	8/5/2016 2:49 PM	File folder
LogicalComplexJoin	8/5/2016 2:49 PM	File folder
LogicalTable	8/5/2016 2:49 PM	File folder
LogicalTableSource	8/5/2016 2:49 PM	File folder
PhysicalDisplayFolder	8/5/2016 2:49 PM	File folder
PhysicalTable	8/5/2016 2:49 PM	File folder
PresentationCatalog	8/5/2016 2:49 PM	File folder
PresentationHierarchy	8/5/2016 2:49 PM	File folder
PresentationTable	8/5/2016 2:49 PM	File folder
Schema	8/5/2016 4:43 PM	File folder
Variable	8/5/2016 2:49 PM	File folder

Figure 6. Structure of folders for a MDS XML Documents project

What these folders specify can be viewed basically by reflecting on the name. Files inside these folders are named by an element id having as naming convention element\_id.xml. Inside these XML folders, are specified in XML tags (unfortunately undocumented or unshared documentation by Oracle) the specific information required to fully define each element.

What is also unfortunate and which seem such a strange behaviour, is that the MDS XML documents created in such way from the Administration Tool cannot be imported back in the Administration Tool itself (at least for the 11.1.1.9.0 which is the last tool Oracle has shared till the moment this paper was written). Following several attempts, I concluded that for some of the files, Oracle does not produce a unique ID. This duplication of IDs fails to open the MDS XML files and the opening of the whole project. Anyway, this problem might be bypassed in case these files are created through some other script, and are not directly created through the Administration

Tool itself. At least this solution might be convenient if the conversion from BO universe to OBIEE repository is done once and the maintenance of the repository is done through the Administration Tool always utilizing the compiled repository (.rpt file).

## 5 BusinessObjects Model

Business Object also follows a very wide variety of integration and development kits in order to allow users an automated access. As per SAP documentation site there are provided “developer guides, API reference material, sample code, and object model diagrams for Java, .NET, COM, and Flex APIs available in SAP BusinessObjects BI 4.x”[Leo16].

The simplest way which might be used to document and develop, without using the Business Object Universe Designer would be to take in consideration the Excel VBA library provided for BusinessObjects. This library is called BusinessObject Designer 12.0 Object Library and can be download in the internet but which is also installed during the Designer installation on a Windows OS running machine.

The usage of this tool is based on two major objects called Designer.Application and Designer.Universe which execute methods in order to get and set specific elements for the Designer. This tool requires authentication to BusinessObjects in order to proceed with any action, making the access really secure.

For sampling on using these SDKs for Business Objects, several online sites can be consulted. [Hil10]

## 6 Migration of the universes to repositories and of the repositories to universes

BusinessObjects allow the users to access and modify information on a Universe, so it is possible to define a new BO Universe starting from a pre-extracted OBIEE repository, which should be simple to read, as suggested by me in the previous chapter a MDS XML documents structure, which might be read from any of the third generation languages of which SAP provides the SDK, and then the data fetched might be used to create the required repository.

From the other side on, these tools (libraries) provided for BusinessObjects, allow us to document a whole universe, in such a detailed way that the information might be used to recreate the same metadata structure which was previously stored inside the Universe itself.

A third generation development language might be used, I propose a language with a strong XML

support, to take this information which might have been stored in files, or even directly through the SDK, and after compiling a MDS XML Documents project, to upload this one for creating a compiled repository which might be used for further modifications from OBIEE Administration Tool (since the export of the Administration Tool in MDS XML does not allow the re-uploading of this document, without recreation and replacement of the IDs as described in a previous paragraph).

## 7 Conclusion

It is possible to build a tool which would facilitate the migration between the two Business Intelligence solutions, BusinessObjects and Oracle BI. One side, the BusinessObjects side is fully automated and functional, while the Oracle BI part still has some small problems while producing the MDS XML Documents project directly from the Administration Tool. This problem makes it simpler for users to switch from BusinessObjects to Oracle BI, while for the other way, IDs of XML elements and folders have to be recreated before usage.

The problem with exporting the MDS XML documents, problem with the ID uniqueness which is very important in XML elements identification, also makes it difficult for OBIEE repository creators to export and re-import the documents themselves. This means that when producing a repository from a universe, all migrations should complete in one hand, without making further transformations on the repository itself, since this change repository could not be exported further more, making all remaining changes subject to manual transformations.

Even though we mentioned it that one of the ways is simpler, the other way is not impossible. Just the creation of the universe from the repository should ignore the existence of the IDs created by OBIEE and regenerate IDs in order to determine elements. IDs would not be a major problem since the identification of the different elements can be done also by other attributes (ex. the name of the element).

The automation tools have to be developed in technologies which should from one side interact with the SDKs BusinessObjects provide (for some of the most popular third generation languages as Java, .Net etc.) and from the other side should have capabilities of simply validating and interpreting XML documents and writing XML documents through specialized functions. These XML elements should be linked

through their IDs sometimes giving name to elements inside the documents hierarchy folders.

These automation tools would allow the users to switch between these Business Intelligence solutions, taking in consideration other aspects for the choice than the migration of metadata between the two solutions.

So, little to no efforts it would be needed in a one to one migration, in case when the migration itself wouldn't require the redesigning or the fine-tuning of the solution.

## References

- [Tur11] Turban E., Sharda R., Delen D., King D. *Business Intelligence – A managerial approach*, Second Edition, 2011, pp. 327.
- [Ora13] *Oracle Business Intelligence Suite Enterprise Edition 11g - Data Sheet*. (2013). Retrieved from <http://www.oracle.com/us/bi-enterprise-edition-plus-ds-078848.pdf>
- [Wha09] *What is a BusinessObjects Universe*. (2009, March). Retrieved from <http://alteksolutions.com/wp/index.php/2009/03/what-is-a-business-objects-universe/>
- [Com] *Compass Information Science*. <http://www.wcompass.com/SupplyProduct.aspx>
- [Ger16] *Gerardnico*. (2016, June 15). Retrieved from [http://gerardnico.com/wiki/dat/obiee/bi\\_server](http://gerardnico.com/wiki/dat/obiee/bi_server)
- [Ora12] *Oracle Financial Services Software Limited, Oracle FLEXCUBE Universal Banking® 12.0 OBIEE Repository Development Guide*, Release 1.0, May 2012
- [Leo16] Leong, K., Moitel F. (2016, January 6). *SAP BusinessObjects BI 4.x - Developer SDK Library*. Retrieved from <http://scn.sap.com/docs/DOC-27465>
- [Hil10] Hilton, D., Stone, A. (2010, May 10). *NET RAS SDK Samples*. Retrieved from <https://wiki.scn.sap.com/wiki/display/BOBJ/>
- [How12] Howson C., Newbould E. *SAP BusinessObjects BI 4.0 The Complete Reference*, Third Edition, McGraw Hill Professional, 2012, pp. 551.