Callisto: Mergers Without Pain

Huong Morris[1,*], Hui Liao[2], Sriram Padmanabhan[2], Sriram Srinivasan[2], Eugene Kawamoto[3,**], Phay Lau[4,**], Jing Shan[5,**], and Ryan Wisnesky[6,**]

IBM T. J. Watson Research, 19 Skyline Drive, Hawthorne,
New York, 10532, USA
[1] IBM Research
thm@us.ibm.com
[2] IBM Data Management
{huiliao,srp,sriram}@us.ibm.com
[3] University of California, Berkeley
kawamoto@haas.berkeley.edu
[4] San Jose State University
ptaclau@gmail.com
[5] Northeastern University
jshan@ccs.neu.edu
[6] Stanford University
wisnesky@stanford.edu

Abstract. As value networks evolve, we observe the phenomenon of businesses consolidating through mergers and businesses disaggregating and then virtually "re-merging" dynamically to respond to new opportunities. But these constituent businesses were not built in any standard way, and neither were their IT systems. An example in the industrial sector is the need to merge product and parts catalogs, and selectively share customer data. Companies that merge can spend a year integrating their catalogs, by which its time for the next deal. As such, business object integration has become a key aspect of today's enterprise. In this paper we describe an innovation where, by integrating product data management (PDM) systems that manage business objects into Extract-Transform-Load (ETL) technology, we can provide a novel cross-industry solution which can be used in a variety of industries.

Keywords: Business Intelligence, Data warehouse, Master Data Integration, Business Objects Integration, ETL Technologies, Enterprise Information Integration.

1 Introduction

As business systems have consistently expanded throughout the enterprise, the need for multiple systems with different architectures to inter-operate becomes ever more

[*] This work was carried out when the author was at the IBM Almaden Research Center, California, USA.
[**] This work was carried out when the authors were at the IBM Almaden Research Center as Extreme Blue interns.

C. Bussler et al. (Eds.): BIRTE 2006, LNCS 4365, pp. 92–105, 2007.

important. To make this happen, enterprises are turning to higher level software, such as the solutions provided by SAP, Oracle's PeopleSoft, Siebel, and the IBM Websphere Product Center, to manage their *business objects* directly. Existing work in this area mainly focuses on how to design business objects in business systems and how to use them in business processes [3, 4, 10, and 11]. The goal of efficient management of distributed information has become progressively more difficult because of increasing heterogeneity and rapid churn. As cited in [5], the major trends of Enterprise Information Integration (EII) and Enterprise Application Integration (EAI) are overlapping and creating a further integration problem.

Every important entity in a business can be represented as a business object. Business objects are then composed into schema models that capture the semantics of business concepts and are directly useful for business processes. They represent the key concepts that a business needs to operate such as people, services, and whatever is sold. Business objects are used directly by business developers to implement business functions. Examples of commercially available product data management (PDM) systems that manage business objects are already mentioned above. However, because of their intrinsic complexity and the fact that they come from different vendors, such products do not interoperate with each other. Most PDM systems represent business objects in terms of a master data management (MDM) system and store them in a backend data store using a relational database. Any techniques for transformation must be able to access these backend data stores. But naïve access to these objects may have unintended consequences, caused by the semantics of the data and the relationships and constraints required of the data. As a result, conventional approaches to integration may not provide an adequate solution.

Our project, Callisto, originated as an IBM Extreme Blue project http://www.ibm.com/extremeblue aimed to study master data integration requirements and develop a prototype information integration system that leverages current data management technology. We will illustrate the approach with meaningful examples drawn from the retail industry. In this paper, we start with business entities that are represented by hierarchical or multidimensional objects that are in turn supported by products such as SAP, IBM WPC, etc. We will show how they can be managed or analysed by application toolsets such as IBM DWE, Microsoft SQL Server Integration Services (SSIS), Oracle Workflow Builder, etc. that understand relational data and business intelligence. By combining these two product sets in this novel way we can provide an integration framework to do management, mining and analytics on disparate master data systems.

2 Why Bring Business Objects into ETL Tools?

Accessing disparate business objects can be complex due to business objects being compound versions of the data embedded inside many databases and unstructured data sources. Depending on how they are represented and derived, they can suffer from problems of inconsistency and misinterpretation. Business functions are about managing business objects and, while based on typical and existing system functions, they therefore suffer the same fate. Such a separation of business concepts and system implementation in theory enables a business developer to construct business components without detailed knowledge of underlying software technologies, such as specific programming languages, communication protocols and database systems, etc. Traditionally, each business process is built as an application system, which encapsulates the specific task. When business objects are embedded in applications, it is dangerous to inter-operate without a deep understanding of the semantics of each of the applications, and this understanding often requires an understanding of the implementation. This makes such business process integration difficult [5].

Integration is usually tackled using one of four main techniques: transformation tools such as Extraction-Transformation-Loading (ETL), replication, database gateways, and virtual data federation. ETL tools are pieces of software responsible for the extraction of data from several sources, cleansing the data, and customized insertion of the data into a data warehouse.

Today's commercially available ETL products are exemplified by the IBM SQL Warehousing (SQW) component within the IBM Data Warehouse Edition (DWE) product, the Oracle Workflow Builder (OWB) within the Oracle Warehouse Management product, and the Microsoft SQL Server Integration Services (SSIS) within the SQL Server 2005 product. These products are able to support data provisioning processes that exhibit complex data flow. Examples of previous ETL work that has focused on the modelling and managing of the ETL processes can be found in [3].

The user of ETL tools can focus on the semantic mapping from a data source to a data target and then let the ETL tool take care of the underlying transformation details. But current ETL technology only supports the lower level software data (e.g. data inside a DBMS). In short, there is an "impedance mismatch" between business-object-aware software (e.g. SAP, IBM WPC) and business intelligence toolsets (e.g. IBM DWE) which provide real time decision support systems for an enterprise. To keep the business objects encapsulated and consistent, and to make business objects transformation easier and smoother, there is an opportunity to use these two toolsets in a more synergetic way. That is the main idea of Callisto.

3 Callisto: Motivation and Challenges

In this section, we describe more of the motivation behind our Callisto project and show some of the challenges that face a business object integration project using ETL technology.

3.1 WPC Business Objects

The Websphere Product Center (WPC) is a product information management (PIM) system that provides a centralized repository for an enterprise's master data. Like most other business objects systems, e.g. SAP, Oracle PeopleSoft, this information is maintained in a relational database in the back end, but is presented to the user as business objects in a way that is rich enough to support a realistic business environment such as may be found in a retail business: *Items* belong to *Categories*, and Categories are organized into *Hierarchies*. Hierarchies are an especially useful feature of the WPC, and are a good example of how business objects simplify user interaction: a user can organize and view the categories in different hierarchies and can create different catalogs for the same item set. A complete overview of WPC schema model will be given later in section 5.2.

We chose to use the Websphere Product Center as our proof of concept project to test the idea of using ETL tools to accelerate integration. Consider the following scenarios:

1. *Business intelligence:* Because the WPC stores valuable master data, we would like to be able to use this information for analysis. For instance, combining transactional data with categorical information from the WPC could allow for analysis of optimal item categorizations. ETL technology is advantageous for this type of scenario because it provides a way to link into business intelligence tools: once information is exposed to the ETL through Callisto, the ETL itself can then be used to manipulate the data to place it into a data warehouse for analysis.
2. *Trading partner collaboration:* Trading partners wishing to collaborate need access to the other partner's information, but if one partner is using WPC and the other partner a relational database, there is no easy way to transfer information between the two systems. By using Callisto, the business using the WPC can expose its business objects selectively, and in a way the other business can use, and the ETL itself can be used to massage this relational data so that it can be exchanged easily.
3. *Global data synchronization:* It is common for businesses to use a single WPC instance as a master information repository and also have redundant information stored in relational systems throughout the enterprise. To synchronize the WPC and such systems, it is necessary to expose the business objects in an ETL tool so that they can be mapped to schemas suitable for use by the other systems (and vice versa).
4. *Catalog construction:* Catalog construction (including catalog merging) is similar to data synchronization in the sense that the initial construction of a WPC instance requires the use of other various types of datastores. To speed up the deployment of a new catalog, Callisto could be used to extract, rather than synchronize, information from various data sources.

3.2 Integration Challenges

Current ETL technology supports relational formats, such as relational database tables, and some file system formats, such as CSV files, etc. To represent business objects inside of the ETL, we must find a way to describe business objects in a relational format without resorting to examining how the business objects are implemented and stored. In essence, we must create custom ETL *operators* that expose the required information. This is not an easy task because business objects are often semi-structured, as in the case of WPC. The following are some key challenges:

1. The relational presentation of a business object must be as rich as the original object. That is, information about the business object should not be lost when the object is represented in a relational way. In addition, the information presented in the relational view must be presented in a way that is useful.

2. In many business-oriented systems, there is no clear boundary between data and metadata. An ETL system requires operators to expose metadata while a dataflow is designed, and to manipulate the data during runtime.
3. Different business objects of the same type may not share common properties, so that there is not necessarily a common relational representation for different instances of a type of business object. For instance, a retail Category business object may be represented as a table with columns for 'name' and 'price', but another Category object may require 'UPC' and 'description. However, both are called Category objects, and so we cannot always decide on relational representations for an entire class of such objects.
4. Business objects and their relational views must relate to each other in a consistent, complete, and useful way. For instance, it is common for one business object to reference another; say, for a person object to reference a department object, thus capturing the relationship that the person is employed by the department. Thus, when multiple business objects are represented in multiple relational tables, if one objects references another, that information must be suitably and consistently encoded wherever it is represented in the relational tables.

4 Illustrative Scenarios and Use Cases

In this section, we describe several use cases that make clear the requirements for integration of WPC business objects using an ETL toolset. A procedure to execute each use case is also given here for illustration. Note that the solution for each use case is very similar to any regular ETL based solution. The difference is that instead of using typical ETL operators such as *join, union, etc.* Callisto's solution uses its own customized operators for WPC business objects. Some of the value will become clearer in later sections when we describe the Callisto internal design and run time engine.

4.1 Use Case 1

Our first scenario will focus on master data integration; a typical customer pain-point of most master data management systems. WPC catalog building is a semi-automatic process that can require a substantial amount of skill and manpower to deploy. This process could be more complex and involved when there is a need for master data integration. In this scenario Callisto's aim is to see whether the process integration of new data into the WPC master catalog can be simplified.

Situation: In a fictitious example, WorldMart, the world's largest retail chain, plans to expand its product portfolio by acquiring HomeMart. The acquisition needs to be completed by integrating WorldMart's product catalog with HomeMart's various data sources.

System Environment: WorldMart uses the IBM Websphere Product Center to centrally manage its product catalog information. HomeMart's product, suppliers, stores, and pricing information are scattered throughout different systems and suppliers' databases.

Limitation: WorldMart's upper management has required that the integration of HomeMart's product information into WorldMart be completed in three months. However WorldMart's systems group estimates that this could take much longer if they have to use 'import' and 'export' functions, such as those found in WPC, for different HomeMart's product data silos. Instead we will develop *'Item Import'* and *'Item Export'* ETL operators that will be defined in more detail in the next section.

Solution using Callisto

1. Define data source: HomeMart's product information using Callisto's operator '*Item Export'*
2. Define target source: WorldMart's WPC product catalog using Callisto's operator "*Item Import*".
3. Connect date source and target using metadata mapping.
4. Complete and dynamically execute Callisto dataflow.
5. Load Data into WorldMart's Master Catalog.

4.2 Use Case 2

Our second scenario shows how Callisto can allow BI queries to be entered against the master data that is found in WPC. Traditionally, BI tools were limited to analyzing transactional data only and business systems such as WPC do not support business intelligence.

Situation: A fictitious online bookseller, Book4Sale.com, found that the sales of its Harry Potter books grew two-fold when it changed the category from "Children" to "Fantasy". Changing categories can dramatically boost sales. Book4Sale.com would like to analyze the past trends of its sales to determine the optimal category for its products.

System Environment: The bookseller uses the IBM Websphere Product Center to centrally manage its products and categories. WPC manages Book4Sale catalog but it does not have the ability of a Business Intelligence tool such as reporting and analysis for evaluating product trends.

Limitation: Conventional data export techniques are too slow to react to the high volume of daily catalog changes. Failing to spot a bad branding or categorization can lead to a huge loss. In addition, peak sales trends may be missed by slow, conventional techniques.

Solution using Callisto

1. Define data source using Callisto's operator '*Item Sources'*
2. Add the ETL existing operator "*Current Time'*
3. Define BI using Callisto's operator '*SCD'* (a.k.a. '*Slow Changing Dimension').*
4. Complete and execute Callisto dataflow
5. Load Master Data into ETL toolset like SQW.
6. Analyze Results using any reporting and analysis experiment, we use IBM Alphablox and data mining tools.

5 Callisto Architecture and Implementation

5.1 Architecture

Callisto, as depicted in Figure 1, is essentially implemented as a set of plug-ins around an ETL system. Our implementation used Eclipse plug-ins to the IBM Data warehouse Edition (DWE) toolset called SQL Warehousing (SQW)[1]. Callisto extracts and loads information into the WPC using the scripting mechanism and a JSP interface. Callisto also transforms information to and from the hierarchical format that the WPC uses by examining a model of WPC business objects. Finally, Callisto presents and receives relational representations of WPC information from the ETL tools set. The ETL tools set provide support for transforming relational information and connectivity to various relational systems, thus allowing WPC information to be integrated into the ETL tools set along with other BI and transformational operators.

We developed a set of *'customized'* WPC operators within the SQW design studio, which represent the business objects for Import and Export functions in WPC:

1. **Item Export:** Export a category of items from the WPC. The items contain the values of their attributes as columns in a table.
2. **Item Import:** Import a category of items into the WPC, with attribute information.
3. **Hierarchy Export:** Export a hierarchy from the WPC, where parent-child relationships are maintained using paths and parent/child columns.
4. **Hierarchy Import:** Import a hierarchy into the WPC, while maintaining parent/child relationships.

Users may use the Callisto operators in the same way that they would use typical ETL operators within the SQW toolset. As such, the DWE and Callisto infrastructure provides a unified framework to do mining and analytics on WPC master data that otherwise would not be easily possible.

Callisto operates by analogy with existing DWE toolset. For both use cases in above section 4.1 and 4.2, Callisto first examines a WPC instance and reads relevant catalog schema information. This information is used to dynamically populate an Eclipse Modeling Framework (EMF) model of the WPC instance. Thus, the model provides information that categories are related, hierarchies have these categories, etc. This model is the foundation of the rest of Callisto.

With a model in hand, user can drag and drop the Callisto operators into an ETL dataflow. Depending upon the operator chosen, the user must select different aspects of the WPC instance in order to create the operator. For instance, in Use case 1, when using Item Export, a user would browse a WPC instance and select a particular category of items to export from HomeMart.

---

[1] Like most commercially available ETL toolset, SQW provides a framework called the *Data Flow*. The data flow is an extensible framework that allows users to build data extraction, transformation and load sequences as a flow of '*Operators'*. The SQW also allows the addition of *Custom operator libraries,* where developers can provide operators to represent their own customized processing in the form of *plugins.* [15].

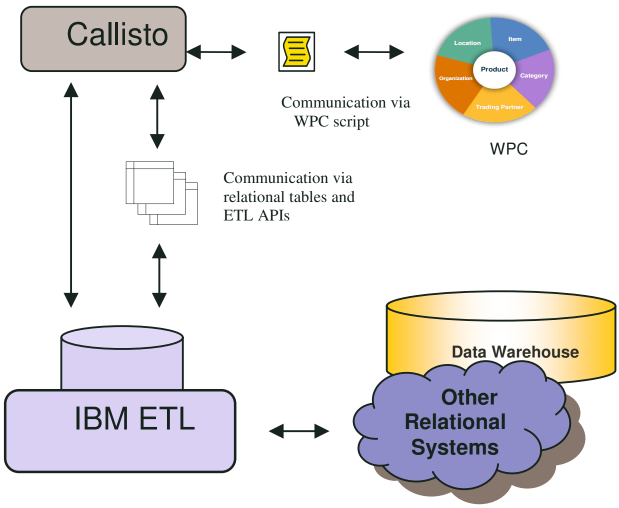

Fig. 1. Callisto Architecture

Callisto provides a code generator for each operator, which generates script that performs the required WPC operation. For example, a *Hierarchy Export* operator causes the generation of a WPC script that involves exporting the whole hierarchy being exported from the source catalog to the target catalog without user's intervention. In Use Case 1, user can select the whole '*Home'* hierarchy within the HomeMart catalog to be export to the target '*Household'* hierarchy in the WorldMart catalog.

Finally, Callisto provides a runtime so that data flows using Callisto operators can be executed. This runtime uses a script-generation technique to create scripts to communicate with a WPC instance, sending or receiving information form as required.

5.2 Callisto MDM Schema Model

Websphere Product Center (WPC) is a product information repository. The "core objects" in the WPC are catalogs, items, attributes, category trees (a.k.a. hierarchies) and categories (a.k.a. hierarchy nodes). *Attributes* hold values or group other attributes. Attributes are defined through specifications (a.k.a. specs).

Items make up the primary data element in WPC. They are typically represented as SKUs, individual products etc. Catalogs are the containers for items. An item belongs to one an only one catalog. Each catalog has one primary specification that defines the attributes that all the items in that catalog share.

Category trees are hierarchical arrangements of categories. This provides users with different "views" into the same set of data. Hierarchies are built and stored separately from items and catalogs. This enables the same hierarchy to be deployed in multiple catalogs, and also allows items in a catalog to be viewed in multiple hierarchies.

Items are mapped to categories. Categories defined specific attributes for the items mapped to them through secondary specifications.

5.3 Callisto UML Data Model

In Callisto, we use IBM Rational Rose [14] to model WPC objects. This UML tool defines data models in a higher abstracted level using a set of well-defined graphical tools. Figure 2 shows the Callisto data model.

Each WPC object is modeled as a standard class. And their containment relationships are modeled as aggregation relationships in the Rose model.

The Eclipse Modeling Framework (EMF) [2] is a Java framework for generating tools and other applications based simple class models. After modeling the above two steps, these models are exported as EMF '*ecore*' (Eclipse modeling framework core) models.

EMF uses these '*ecore'* models and generates customizable Java code that can then be used to manage the life cycle of these business objects, including their relationships as well as provides means of serializing and de-serializing these objects as XML/XMI files.

Note that our UML model of business objects is incomplete and is a simplification; however, it is sufficient and simple for us to use in this prototype. The code generated from the UML and EMF artifacts are what we refer to as the 'WPC model'.

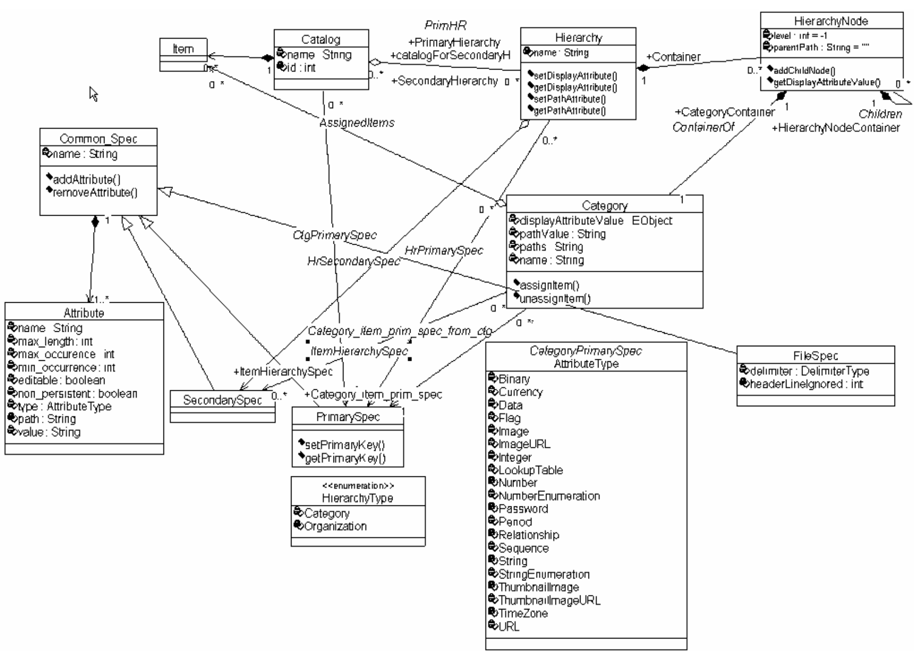

Fig. 2. Callisto MDM Data Model

5.4 Callisto Implementation

Callisto has both a design time and runtime component. We will first describe the runtime in Figure 3.

Figure 3 shows a cloud representing the sources of data that the ETL runtime is capable of connecting to. This can be essentially any form of data, as the runtime is capable of invoking arbitrary Java classes at run time. So as long as an adapter Java class is created, the runtime can connect to anything. The runtime itself currently has adapters for databases that use JDBC, along with a few other special data sources and sinks.

The ETL Data flow code generator creates an execution plan that is woven together from the generated code of the operators connected in the Data flow designer. The Callisto operator code generators generate WPC scripts corresponding to the operator properties that the user set. The ETL runtime takes care of moving the data around in *Virtual Tables* for the runtime code to use.

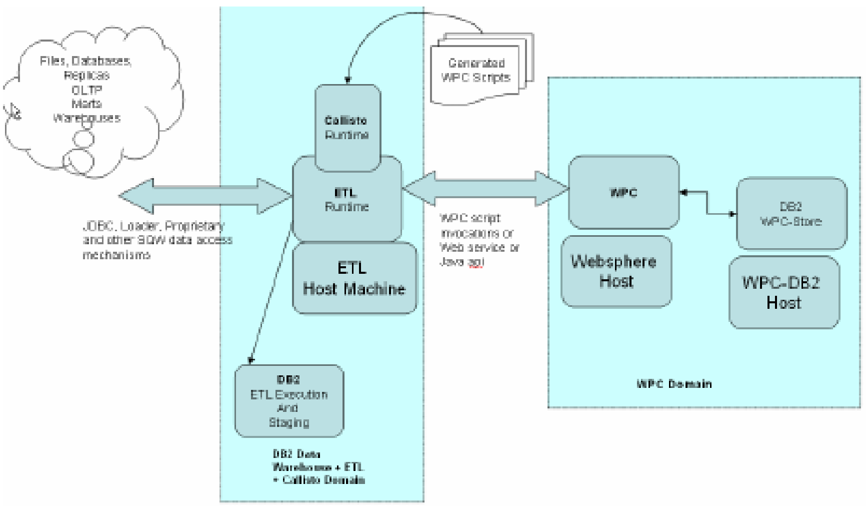

Fig. 3. Callisto runtime interactions

The Callisto component generates WPC scripting code to perform its work; the ETL runtime uses a Java runner [1] to invoke the Callisto runtime Java class which executes the WPC code (and using the generated script as input) remotely against the WPC instance. The ETL runtime can run any executable, including a Java class. WPC script generation is performed at code generation time and these serialized scripts are directly referenced by the generated code. For instance, to export items, Callisto generates a script that pulls all items from a category. Then at runtime, this script is executed via an *'execution plan'* which can be run (or scheduled) using the SQW runtime component.

The runtime component of the ETL takes care of running the execution plan and moving the data around from component to component. In a production environment, it would typically run inside of an IBM Websphere Application Server that is running other business logic applications. For our purposes, it can also be housed inside the same Eclipse instance as the design environment.

Figure 3 also shows some other elements of the Callisto runtime that are now described.

The **Data Warehouse** component represents a generic data warehouse; it may be a database like DB2 with an associated Business intelligence tool and data mining components. Regardless of its form, the data warehouse represents one possible final destination for the data extracted from WPC.

The **WPC** is a J2EE application that we are using as our source (or target of transformed items) of catalog data. The instances of WPC are the web applications running inside of an application server. Connections to the backend WPC storage database will usually occur over a network. WPC information is stored as tables in DB2, however this information is "internal" and only very specific API is allowed to manipulate the WPC catalog. WPC provides a web based interface to present the catalog in a hierarchical format to the end user, as well as provides an administration interface to maintain the product catalog.

WPC uses a **DB2 database** to store its information. A DB2 instance is also used by the ETL runtime engine as its SQL execution database as well as a repository for staging temporary content.

We now describe the Callisto design time that is shown in Figure 4. The elements of Figure 4 are as follows.

The **Eclipse** platform provides a graphical workbench and extendible plug in architecture. The ETL design tools use Eclipse as part of the editing environment. By

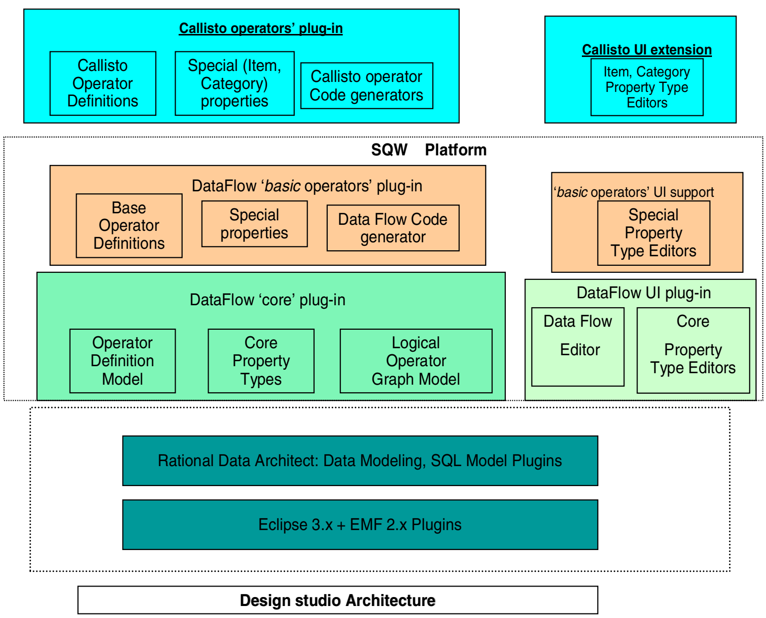

Fig. 4. Callisto design structure

registering Callisto with ETL, the ETL platform becomes aware of its existence and makes it available to the designers of a dataflow, where it can be used like any other Data flow operator.

The **ETL plug-ins** for Eclipse also contains the graphical designer and associated pieces and the Data flow code generator. Once a dataflow is developed, the Data flow code generating system compiles the operators into an Execution Plan Graph that can be executed on the ETL runtime. The ETL runtime thus will be able to invoke the Callisto Java runtime classes with input the WPC scripts that are generated by Callisto's code generators. The Callisto operators appear in the Data flow design tool as regular operators.

Figure 5 shows a screen shot that captures Callisto at run time for the second scenario where on the top right hand section shows the SQW Design Studio area. In this area, several SQW pre-defined operators (*Add Current Time* operator and *Slowly Changing Dimension* operator) and one Callisto operator, *WPC Item Source* operator, are shown. They were selected from the operators' palette at the right hand side of the Design Studio. When the *WPC Item Source* operator is defined or clicked from the palette, SWQ Design Studio, acting on behalf of Callisto, will open up an instance of Book4Sale catalog. From this catalog, which appears in the bottom right hand pane of Figure 3, user selects *Fantasy* category and can see that this category is defined by 2 attributes: *id* and *name*. When the data flow is run, this operator will connect the WPC instance and present this selected item source, namely *Fantasy*, in the WPC category

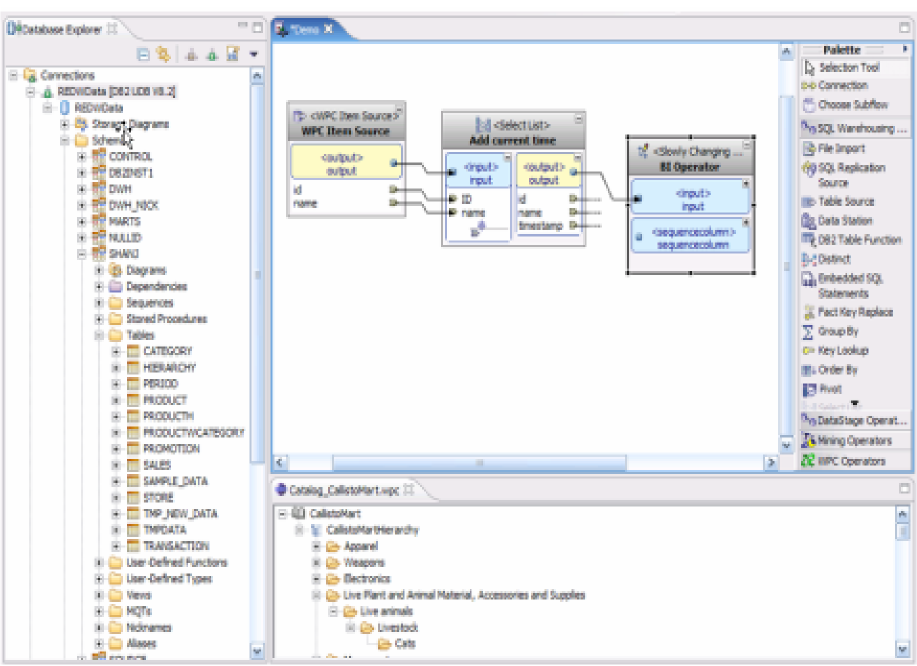

Fig. 5. Callisto screenshot at runtime for Use Case 2

as a relational table to the next SQW operator, *Current Time,* which will then add time stamps information to this selected WPC category. The last SQW operator in the second scenario is the *Slowly Changing Dimension* operator. It takes the input data from previous operators, *WPC Item Source* and *Current Time*, and merges it with existing data in the data warehouse. Once the dataflow is run and complete, the new information will be available for analysis with a reporting tool such as IBM Alphablox.

6 Conclusion

We have demonstrated that by adding some conceptually simple Java-based operators to a transformation tool business objects can be simply integrated, assembled or disassembled.

Our approach has been relatively simple and makes use of commonly available technology: we use UML and EMF modeling, which captures the key constraints between objects, to generate Java code. The Java code is used to present relational representations of selected business object instances based on the object's state. Finally, custom operators use these Java objects to present clean relational table schemas (virtual tables) to the rest of the ETL transformation framework.

While we are only able to provide initial results at this point, we believe there can be some encouragement that existing business intelligence tools can be combined in this novel way to address data residing in relational database systems and therefore simplify and improve some approaches to data integration and business intelligence.

Further experiences and development of more and richer operators for different master data system such as SAP, PeopleSoft and Siebel would make mergers with Callisto within the ETL toolset even more widely applicable.

Acknowledgments. The authors thank the IBM Software Group, especially the Business Intelligence ETL and the Websphere Product Center development groups for providing resources to this project. The authors also thank the anomymous reviewers for a thorough and constructive set of reviews of this paper.

References

1. Apache Velocity Engine http://Jakarta.apche.org/velocity
2. Eclipse Modelling Framework (EMF) http://www.Eclipse.org/emf/
3. Eeles, P., Sims, O.: Building Business Objects. Wiley Computer Publishing, Chichester (1998)
4. Gillibrand, G.: Essential business object design. Communications of the ACM 43(2), 117–119 (2000)
5. Halevy, A., Ashish, N., Bitton, D., Carey, M., Draper, D., Pollock, J., Rosenthal, A., Sikka, V.: Enterprise Information Integration: Successes, Challenges and Controversies, ACM SIGMOD, pp. 778–787 (2005)
6. Madnavan, J., Halevy, A.: Composing Mappings among Data Sources. VLDB, pp. 572–583 (2003)

7. Maier, A., Mitschang, B., Leymann, F., Wolfson, D.: On combining business process integration and ETL technologies. BTW, pp. 533–546 (2005)
8. Morris, H., Lee, S., Shan, E., Zeng, S.: An Information Integration Framework for Product Lifecycle Management of Diverse Data. ACM JCISE, vol. 4(4) (2004)
9. Simitsis, A., Vassiliadis, P., Sellis, T.: Optimizing ETL processes in data warehouses. In: Proceedings of the 21st International Conference on Data Engineering, ICDE, (2005)
10. Sims, O.: Business Objects, Delivering Cooperative Objects for Client-Server. McGraw-Hill Book Co., New York (1994)
11. Sutherland, J.: Business Objects in corporate information systems. ACM Computing Survey 27, 274–276 (1995)
12. Vassiliadis, P., Simitsis, A., Skiadopoulos, S.: Conceptual modelling for ETL processes. DOLAP, pp. 14–21(2002)
13. Websphere Product Center: Architecture and Component Overview, Version 5.1 (August 2004) http://www.ibm.com/software/integration/wpc/documentation
14. IBM Rational Rose, http://www.ibm.com/software/rational
15. DB2 Data Warehouse Edition Version 8.2 http://www-128.ibm.com/developerworks/db2/roadmap/dwe-roadmap-v8.2.html