InDetail

Bloor

REVER

An InDetail paper by Bloor Research
Author : Philip Howard
Publish date : October 2008

...the use of REVER should mean that the task is completed faster, with less expense and with greater assurance about the quality of the migration

Philip Howard

Executive summary

REVER describes its products, which consist of DB-MAIN and various programming language analysers, as being about the development, maintenance, control and evolution of information systems, particularly with respect to the data contained within those information systems. Particular strengths lie in the areas of data migration, legacy application migration and through providing complementary facilities to traditional data quality products. In this InDetail paper we will discuss these various aspects of the company's offering. DB-MAIN can also be used as a conventional modelling tool for the construction of new applications or for data modelling. However, these aspects of the product are not discussed in the paper, where we will be focusing primarily on the use of REVER for migration projects, though we also briefly consider the complementary nature of its use for data quality purposes.

Fast facts

REVER is not just based on modeling techniques but employs a formal model-driven architecture. That is, it engineers (both forward and reverse) data in terms of a hierarchy of conceptual models, logical models and physical models. This is important because all databases share a common conceptual model and all databases of the same type share a common logical model. Thus DB2, for example, shares a common conceptual model with, say, CA-IDMS, and a common logical model with Oracle or SQL Server. What this means is that you can migrate from any database implementation to any other by simply mapping up this physical-logical-conceptual tree as you far as you need to (the point at which the models are common) and then mapping down again.

It is worth noting that the model-driven architecture (MDA) approach that has been adopted by REVER is a formalised methodology that has been standardised by the Object Management Group. It is quite widely used for application development, though it is less common elsewhere.

Key findings

In the opinion of Bloor Research the following represent the key facts of which prospective users should be aware:

- REVER employs a Model-Driven Architecture to support the development and migration of information systems with specific emphasis on data.
- REVER provides support for most of the popular modern and legacy programming languages (as well as 4GLs) so that implicit data rules built into applications can be discovered.
- Best practice in data migration is to do this by migrating business entities rather than working merely at the table level (when relationships can get broken). REVER supports this.
- REVER only makes its software available via systems integrators (via knowledge transfer, as a black box service or in partnership). We would like to see the company making its tools available directly to end users requiring it.
- REVER is currently only available in France, Benelux, the Netherlands, the Nordic countries and the USA.
- REVER can be used in complementary fashion to data quality tools because it discovers how data is used by different applications and therefore provides a context to that data.

The bottom line

The principles behind REVER are not new: reverse engineering and model-driven architectures have been around for a number of years. However, the two have not previously (as far as we know) been combined within a data-centric, as opposed to an application-centric, environment. The result is that REVER goes much further in automating the discovery of relevant structures than more conventional approaches. While you will still need domain experts to make relevant input into a migration process this should not be nearly as onerous when using REVER as when employing traditional techniques. While the product's availability is limited by geography and the company's sales model it should certainly be seriously considered by any companies falling within these boundaries and, particularly, for migrations from legacy environments.

The product

The principle lying behind REVER is that it is model-driven. Briefly, this means that it treats information in terms of:

- A Conceptual Model, which is a semantic description of the information system, independent of how the technology is implemented.
- A Logical Model, which is a technical description of the information system related to a particular technology.
- A Physical model, which is a technical description of the information system related to a specific implementation.

Here is an example of how this works in practice: all databases share a common conceptual model, relational databases share a common logical model and the physical model represents the actual implementation of any particular database. In other words, these represent succeeding levels of abstraction that can be used to map (and therefore migrate) from, say, a network database such as CA-IDMS to a relational database like Oracle. More generally, application developers usually work at the level of the logical model whereas database administrators are typically concerned with the physical model.

In terms of specifics, REVER applies this model-driven approach both to the construction of new systems (model-driven engineering) and the capture of information about existing systems (model-driven reverse engineering). Further, as is illustrated in Figure 1, this is applied at either the data level (MDDE) or application/process level (MDPE). Essentially, DB-MAIN provides the necessary capabilities at the data level while the company's analyzers work in conjunction with DB-MAIN at the process level.

Product technical details

DB-MAIN is a modeling tool that does the normal things that you would expect a data modeling product to do (using both UML and Entity-Relational based modeling techniques), but which, more specifically, is used to define and store the transformations that are needed to accurately transform between physical and logical models and between logical and conceptual models, and vice versa; in both cases without any loss of semantic integrity.

DB-MAIN was first released in 1995 and is currently in version 8.02 (version 9.0 will be released by the end of 2008). Databases supported include IMS, DL1, CA-IDMS, CA-Datacom, IDS2, Supra, Sybase, Adabas, DB2, Oracle, PostgreSQL and SQL Server amongst others. The company also offers language analyzers for COBOL, C, C++, Java, Visual Basic, PL1, Mantis, Natural, CoolGen and NatStar as well as various data integration tools from vendors such as Informatica, IBM, SAS and Hummingbird (OpenText). Note that this last means that you can use REVER to migrate from one ETL (extract, transform and load) tool to another.

The software supplied by REVER runs on Windows (the next release will also run on UNIX/Linux).

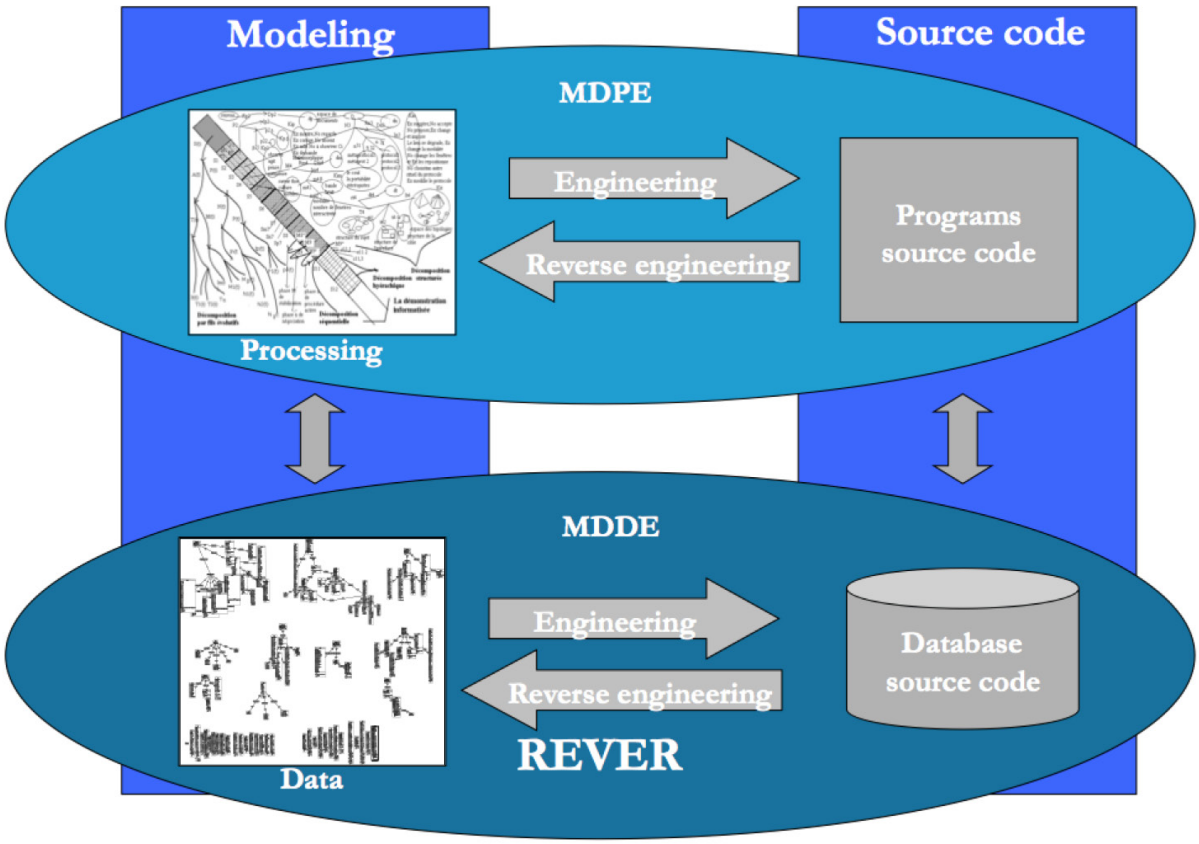

Figure 1: REVER Model Driven Engineering (MDE)

The product

Product description

REVER is suitable for most types of data migration but not all. Specifically, the exceptions would be environments where you are migrating from one version of an application environment, such as SAP R/3, to a later version. Similarly, if you are consolidating multiple instances of an application such as Siebel then REVER is probably not the best approach to take. This is not to say that REVER could not cope with these environments but that it lacks the in-depth knowledge of these application environments that are available elsewhere. Conversely, where REVER has particular strengths is in supporting migration across environments.

The cornerstone of REVER is its reverse engineering, which is illustrated, along with its deliverables, in Figure 2.

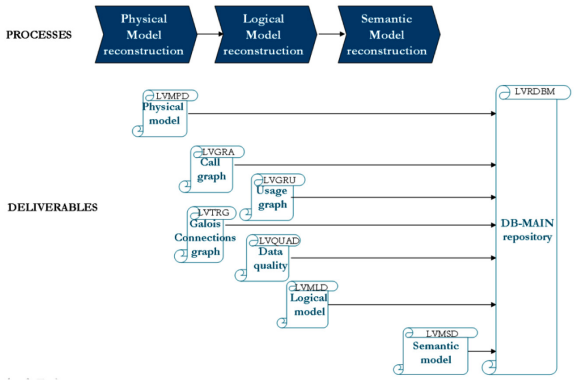

Figure 2: Reverse engineering in REVER

For the physical model reconstruction, REVER analyses the relevant DDL to discover all of the entities involved, along with their attributes and the relationships between entities, with relevant details being stored in the DB-MAIN repository. To this extent REVER is by no means unusual. However, the results of such reverse engineering are typically unrefined and where REVER adds capability is in its additional analyzers that will, for example, read COBOL copybooks in order to gather additional details regarding the physical model so that the complete discovery of the physical model, with all of its structures and relationships, can be completed.

Reconstructing the logical model requires a number of further steps. The first thing that has to be done is to strip out anything (for example, indexes) that is related to the physical implementation of the model. This is achieved through the application of specialised functions built-in to REVER. Next, procedural source code (database triggers and stored procedures, application programs, JCL and so forth) needs to be analysed in order to discover any implicit data rules that may impact on the model. This is done in two stages, which the company refers to as macroscopic and microscopic analysis, respectively.

During macroscopic analysis the purpose is to identify all the calls that exist between different parts of a program and all the instructions that access the data (SQL requests, DML verbs and so on). This results in two sets of information that can be represented graphically, either separately or together. Figure 3 shows a screenshot depicting a combined call and usage graph, where the former indicates the links between different programs and the latter illustrates the links between procedural objects and data objects.

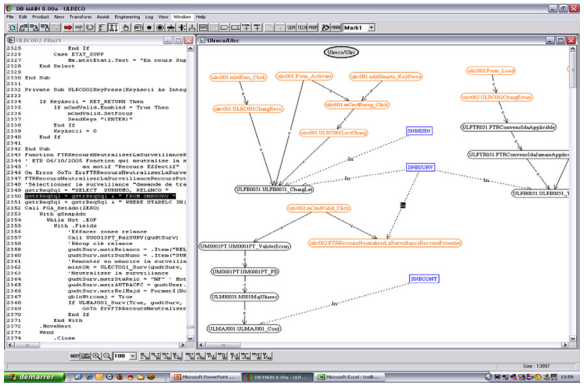

Figure 3: Combined call and usage graph

Note that there are facilities provided so that you can see each procedural object independently or you can look at the whole picture, as required.

An alternative representation of call and usage data is provided by means of a Galois connections chart, an example of which is illustrated in Figure 4. This not only illustrates the relationships between procedures and the tables (data objects) that they access but also (by applying a mathematical technique known as a Galois transform) groups the information (via the spread of the dots) so that you can see sets of data that it would make sense to migrate within a single batch because of the inter-relationships involved.

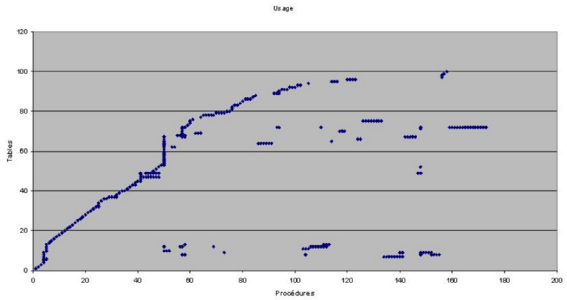

Figure 4: Galois connections chart

The product

Next, there is microscopic analysis, the purpose of which is to discover any data rules that may be hidden in application programs. Here, REVER uses its program analyzers to inspect the relevant source code alongside the partial (following macroscopic analysis) logical model and to identify the data flow within a program and through a chain of programs. The results are illustrated in Figure 5, which shows dependencies within an IDS2 database. Here, data flows are shown in the top pane (Record 2 to Record 1), line numbers in the source code at bottom left and relevant source code extracts in the mid-lower pane.

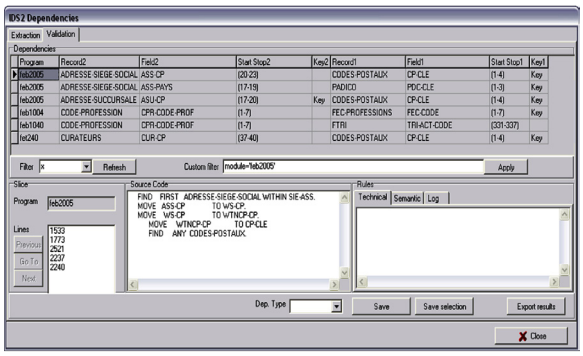

Figure 4: Showing dependencies within an IDS2 database

Data flows need to be inspected by IT staff to determine if they are significant and, where that is the case, these will be automatically integrated within DB-MAIN to enrich the logical model and, at this point, the logical model is complete. However, it is important that this logical model be validated before using it for further development, so the final stage is to conduct a data analysis whereby control requests are generated by REVER from the data rules (referential constraints, attribute types and so on) recorded in the logical model to test the logical model and to identify any data that does not conform to these rules, so that appropriate remedial measures can be applied.

Finally, we come on the reconstruction of the conceptual or semantic model. REVER provides a number of transformation functions; for example, you can decompose compound attributes, integrate entities, suppress records and attributes, and so forth, by means of the product's Transformation Assistant, and there are facilities to do things like automatically determine sort keys (in network databases). All the transformations are traced in order to keep the mappings between all the elements. This mapping can be viewed and modified through the Mapping assistant. However, you will also need knowledge of the application domain to finalise the conceptual model.

Data and database migration

How much of REVER you will use for data migration will depend on what you are migrating between. If you are migrating from one relational database to another then you will not need to create a conceptual schema and your process flow will be to reconstruct the physical model and logical model of the existing database and then map that forward to the new database. Conversely, if you want to migrate from a network database to a relational database then you will have to map up to the conceptual level and then back again. We have not described these forward mappings as REVER is little different from other tools that have this capability.

Data quality

It is useful to consider REVER in terms of data quality as well as data migration and data modeling. This is not because REVER provides data cleansing and similar facilities but because it provides complementary capabilities for data quality that are not provided by traditional data quality vendors. Specifically, REVER discovers where, and by which programs, data is used. In other words, it understands the context in which data is used, whereas conventional data quality products treat data in a stand-alone fashion with regard to usage. It is not hard, for example, to imagine different application programs using the same data in different ways and what may be data cleansing in one context may create dirty data in another.

The vendor

Vendor background

REVER, which is French for dream (as in reverie), is a Belgian company that has taken research originally undertaken at Namur University and commercialised it. The company was founded in 2005, supported by both private and public investors. The company has offices in both Belgium and France and partners in the Netherlands, Nordics and USA.

Although the company offers professional services, it only markets to systems integrators and vendors rather than end users, either on a project basis or as a black box or via technology transfer. Partners in this arena include Bearing Point, CSC, Unisys, Bull and TietoEnator amongst others. Some of the more well-known customers that have used REVER's technology include Chrysler, Mercedes-Benz, the Generali Group and the Belgian ministry of Finances.

REVER web address: www.rever.eu

Summary

Suppose that you want to migrate from an Adabas environment to one using DB2. You would typically use a modelling tool to reverse engineer the Adabas database and to model the new DB2 environment. Then you would use an ETL (extract, transform and load) tool to move and transform the data. However, there is a big gap here: apart from the fact that most ETL tools have no understanding or support for business entities as opposed to tables and the like, neither modelling nor ETL tools have any ability to discover implicit data rules built into applications. There is therefore a large amount of manual work involved in such a process. By automating much of this the use of REVER should mean that the task is completed faster, with less expense and with greater assurance about the quality of the migration.

The difficulty that REVER faces is precisely that it offers a different approach compared to the leading ETL vendors in particular. In such circumstances our experience is that the companies that succeed in building a significant business are those that are as much focused on marketing as on the technical aspects of their offering. This is the challenge that REVER faces—if it can get its marketing right it could become, and deserves to become, a major player in the data migration market—whether it will succeed in doing this remains to be seen.

Further Information

Further information about this subject is available from http://www.bloor-research.com/update/980

Bloor Research overview

Bloor Research has spent the last decade developing what is recognised as Europe's leading independent IT research organisation. With its core research activities underpinning a range of services, from research and consulting to events and publishing, Bloor Research is committed to turning knowledge into client value across all of its products and engagements. Our objectives are:

- Save clients' time by providing comparison and analysis that is clear and succinct.
- Update clients' expertise, enabling them to have a clear understanding of IT issues and facts and validate existing technology strategies.
- Bring an independent perspective, minimising the inherent risks of product selection and decision-making.
- Communicate our visionary perspective of the future of IT.

Founded in 1989, Bloor Research is one of the world's leading IT research, analysis and consultancy organisations—distributing research and analysis to IT user and vendor organisations throughout the world via online subscriptions, tailored research services and consultancy projects.

About the author

Philip Howard
Research Director - Data

Philip started in the computer industry way back in 1973 and has variously worked as a systems analyst, programmer and salesperson, as well as in marketing and product management, for a variety of companies including GEC Marconi, GPT, Philips Data Systems, Raytheon and NCR.

After a quarter of a century of not being his own boss Philip set up what is now P3ST (Wordsmiths) Ltd in 1992 and his first client was Bloor Research (then ButlerBloor), with Philip working for the company as an associate analyst. His relationship with Bloor Research has continued since that time and he is now Research Director. His practice area encompasses anything to do with data and content and he has five further analysts working with him in this area. While maintaining an overview of the whole space Philip himself specialises in databases, data management, data integration, data quality, data federation, master data management, data governance and data warehousing. He also has an interest in event stream/complex event processing.

In addition to the numerous reports Philip has written on behalf of Bloor Research, Philip also contributes regularly to www.IT-Director.com and www.IT-Analysis.com and was previously the editor of both "Application Development News" and "Operating System News" on behalf of Cambridge Market Intelligence (CMI). He has also contributed to various magazines and published a number of reports published by companies such as CMI and The Financial Times.

Away from work, Philip's primary leisure activities are canal boats, skiing, playing Bridge (at which he is a Life Master) and walking the dog.

Copyright & disclaimer

2nd Floor,
145–157 St John Street
LONDON,
EC1V 4PY, United Kingdom

Tel: +44 (0)207 043 9750
Fax: +44 (0)207 043 9748
Web: www.BloorResearch.com
email: info@BloorResearch.com