Model-driven Integration Architecture to Overcome Data Complexity

Why we Need Rational Approaches to Face Miscellaneous Issues

M. Roux¹, and T. Pages²

¹BODATAConsulting, 166 avenue du Maine, 75014 Paris, France
²Department of Structure Design Informatics, SANOFI, 371 rue du Professeur Blayac, 34080 Montpellier, France

Abstract - Integration of data resources is widely used by organizations involved in competitive, high-value-added domains. Various solutions are developed to deal with integration issues even if they lack rational methods to face emerging complexity due to semantic connectedness of data sources, especially in Life Sciences and Health. To get credible and best valuable output, needs and solutions must be formalized as a set of unambiguous statements telling “Which”, “When”, “For what” and “How” data integration is designed and implemented. Model-driven approaches were found to achieve these requirements. In this paper, model-driven data integration was put in perspectives in the context of current approaches with special attention to semantic complexity.

Keywords: Data integration, Model-Driven Engineering, Meta-Model, Domain Model, Semantic Complexity, conceptual model.

1 Introduction

Integration of multiple, remote data resources aims at combining selected systems so that they form a virtual new whole. With this respect, data source integration is of strategic importance to organizations involved in knowledge-based research and economy. Especially, it is difficult to maintain current knowledge of appropriateness information without extensive data selection and management. Unfortunately, data resources are not designed for integration and doing it has raised many difficulties due, notably, to semantic and modeling heterogeneity between systems. Today, billions of distributed data sources are provided on the internet as data publishing system, and constitute an unlimited information resource. The only dark shadow (quite dark) consists in specific problems not only because of data volume but because of complexity due to high semantic connectedness of data resources; by “semantic connectedness”, we mean that a data resource cannot be used standalone to get valuable output but must be contextualized and interrelated to other resource contents, making semantic integration a central issue. With this respect, model-driven engineering (MDE) has provided principles, methods and tools to address mapping concerns between technological spaces, opening new dimensions in data integration approaches.

In previous work, we have discussed reasons for implementing model-driven approaches to represent domain data in high-throughput biology [1]; these approaches were first used to develop metamodeling architectures for complex data integration [2] and further applied to designing and populating a data repository [3]. Virtualization of remote resources operates as another alternative to data integration and we showed that data can be integrated by manipulating data models through ordinary metadata transactions [4].

In this paper, we introduce some new possibilities of model-based data integration rooted on model-driven interoperability advances.

The paper is organized as follows: first, major achievements for integrating distributed sources are reviewed; second, basic concepts and methods in Model-Driven Engineering (MDE) are presented before introducing our model-based approach for tackling data complexity with special attention to semantic complexity. Last, we discuss future work looking forward to pursuing our efforts on worldwide biomedical data sources.

2 Key aspects on virtual data integration

Data integration still is an on-going challenge and multiple reviews are driving the debate [5-9]. Virtual integration is opposed to physical integration in the sense that, in one case, data resources are maintained at the origin and presented as a new, virtual whole; in the other case, data are physically extracted and loaded into a centralized data warehouse.

Over four decades, challenges and achievements in virtual data integration have parallel the development of new forms
and new contents of digital data resources as well as changes and improvements in accessibility. Dominant virtual data integration architectures are as follows:

2.1 Federated architectures

The bulk of the early federated databases (FDBs) literature was concerned with the requirements to federate a collection of heterogeneous, distant databases by examining steps to achieve and data formats, available and/or to be developed. Foundations were provided by [10] that reported five-level architecture to deal with files and structured databases. Above each database, a wrapper [11] or translator [12] were designed to convert a search made by an application/user into one or more commands understandable by the underlying source; conversely, when the wrapper received a result from the source it was converted into a format understood by the search application/user. Example of such a wrapper is given with the Object-Exchange Model (OEM): each value to be exchanged using OEM is assigned a tag to a set of tuples; although these labels are not related to any ontology terms and could even have different meanings in different sources [12].

In parallel, federated data systems (FDSs) were developed to support a wider range of data sources including semi-structured data repositories, digital media, etc., based on using wrappers and mediators. [11-13]. According to [13], mediators were “modules occupying an explicit, active layer between the user’s application and the data source”. They were used to integrate multiple and heterogeneous data resources that deal with the same real-world entities; directly above the wrappers, mediators resolve discrepancies between sources; for example, they might contain rules that connect an input ontology to the database schema.

TSIMMIS [14] is a system that deals with semi-structured and textual data resources; it implements rules to manage how data resources must be combined and integrated.

2.2 Brokering architectures

With creation and management of information brokering architectures, particular attention was drawn on semantics (increasingly domain-specific) and the problem of knowing the contents and structure of information resources took second place. The concepts of federated databases were adapted and extended through the creation and administration of various forms of metadata and ontologies [15, 16]. Thus, brokers are exchange devices that take requests from users, translate in terms of some ontologies and dispatch requests to the relevant referenced services; in return, they merge and display the results from the services. Brokers are used in the context of the internet; for example, the Global Earth Observation System of Systems (GEOSS) has developed a broker framework [17], which affords mediation and distribution functionalities to interconnect distributed and heterogeneous resources. This is characteristics of a System of Systems (SoS) environment specially designed for multidisciplinary communities [18]. Thus, GEOSS allows bridging communities without asking them neither to adapt to one single conceptualization of the world nor to change their way of working.

3 Model-driven Data Integration

Main considerations to examine soundness of Model-Driven approaches to data integration are relying on the existing theoretical basis for specifying: (i) model design, i.e. “what” the model is representing, (ii) model properties and constraints, i.e. “how” model building blocks are arranged (syntax and semantics) and model handling i.e. “which” treatments (mapping, merging, etc.), are going to be applied; all of these three intents being in line with model-driven concepts and methods are also corresponding to virtual data integration issues as reviewed above.

3.1 Highlights on Model-Driven Engineering

Model-Driven Engineering (MDE) has emerged and matured in the field of software development. Approaches are built on the core principle that models are first class citizens. There are numerous definitions for the concept of model but we adopted the following “a model is an abstraction of a system built with an intended goal in mind”, lay down by Bézivin and Gerbé [19]. This statement allows identifying the relation isRepresentedBy (µ) linking the system under study to its corresponding model. More precisely, the model can either describe or specify the system and the differences between meanings are attributable to which was built in connection with the other; for example, a system isRepresentedBy (isSpecifiedBy) a model will mean that the model will give value to the system; conversely, telling a system isRepresentedBy (isSpecifiedBy) a model will mean that the system will give value to the model [20].

Building blocks for modelling are provided by the metamodels which are at the heart of MDE. A metamodel is described as “a model that defines the language for expressing a model” [21]; it is a graph of concepts and relations between these concepts. There are a number of languages for writing metamodels like MOF at OMG or ECORE at ECLIPSE.

A model derived from one metamodel shares the metamodel properties and constraints and isConformTo (γ) its metamodel. Thus, several models ConformTo one metamodel will share properties and constraints with each others.
In the light of the above, MDE is based on a four-level structure with defined steps:

- Level M0 corresponds to the part of real world under investigation or the system (physical or abstract) of interest;
- Level M1 is the model level and represents the system at level M0;
- Level M2 corresponds to the meta-model that delineates a set of concepts and relations between concepts and provide building blocks for domain modeling;
- Level M3 defines the meta-meta-model that is “the model that defines the language for expressing any metamodel” [21].

The reification of the notion of model has led to the definition of model properties and operations in which models take part, especially transformations which are central to MDE. Metamodel-based transformations permit descriptions of mappings between models created using different metamodels, and different technological spaces. Practically, transformation rules are designed at the metamodel level between source and target metamodels (transformation metamodel) and executed at the model level [20]. Languages for model transformations are mainly QVT (principally for software development) and ATL (dedicated to solving data engineering transformation problems) [22]. The open source Eclipse platform provides MDE community tools in the context of the Modeling Project (http://www.eclipse.org/modeling/).

3.2 Metamodeling as a rational approach to data integration

Traditional architectures for data integration were adding a rough middle layer to create the necessary bridge between data sources and user layers. Although it is easy to agree on these principles, all the characteristics of these architectures are not readily stated and understood especially given that it is impossible to perform data integration by mapping all the data to one single model (that would force users to adapt to one single view of the world). It is just inevitable to accept the diversity of systems within different business domains and scientific communities. Thus, various types of views are developed and implemented, leading to more complexity over actual complexity.

Abstraction is a well known alternative approach to deal with complexity and the abstract metamodel level could provide building blocks for addressing syntactic, schematic, and structural issues in addition to the problem of semantic heterogeneity. Furthermore, metamodeling affords the method for specifying consistency of the various architecture artifacts on different layers and in different views. Domain Specific languages (DSL), which are specific metamodels provide similar functionalities in line with the variety and the complexity of domains under study.

Thus, rather than addressing data integration issues at the schemas level that makes each data source a particular issue, the problems could be considered at a further level of abstraction to rely on MDE methods and tools.

3.3 Unravelling high-level heterogeneity

To emphasize the interests in implementing Model-Driven approaches, syntactic and semantic data integration were addressed independently as a way to deepen understanding how each approach contribute to the whole data integration process.

3.3.1 Model-Driven approach to syntactic heterogeneity

Syntactic heterogeneity concerns differences in representation format; for example, relational and Entity-Relation formats are used in most of databases. Curiously, the notion of metamodel has emerged since first database management systems (DBMSs) to support schema management. In DBMSs, metamodel is named catalogue or meta-base. Thus, the metamodel below (Figure 1) is currently used for the specification of relational database schemas; accordingly, each new relational model behaves as an instance of this metamodel.

```
BASE(BaseName, BaseId, RelatNbr, Volume,...)
RELATION (RelName, RelId, BaseId, AttNbr,...)
ATTRIBUTE (AttName, AttId, RelName, BaseId, Type, Long,...)
CONSTRAINTS (ConstName, ConstId, BaseId, ConstText,...)
RIGTH(RelName, BaseId, LoginName, RightList,...).
```

Figure 1. Metamodel constructs for the specification of relational database schemas.

Similarly, metamodels for various formats were made available. The metamodel for Entity-Relation is notably used in computer-aided software engineering (CASE) tools; the well known XML schema description (XSD) corresponds to the metamodel of the XML format. One important consequence in implementing a metamodel approach is the possibility of using MDE methods. Thus, object format used for database design can be transformed into ER format for database implementation and populating. Thereafter, data stored in ER format can be transformed to XML format and merged within XML databases. But that doesn’t include structural and semantic heterogeneity that deal with differences on schema (for example, differences in attributes of two schemas) and meaning (for example, the use of synonyms to express the same idea, concept).
3.3.2 Model-Driven approach to structural and semantic heterogeneity

Structural heterogeneity consists in the use of different building blocks to express the same idea. For example, the type of unit of measurement in biological sciences was represented as a relation (Unit in Figure 2.a) or a class (Unit in Figure 2.b) in FuGE-OM [23] and MAGE-OM [24], respectively.

![Figure 2](image-url)

Figure 2. Structural differences between measurement modelling in FuGE and MAGE data models (see below for references).

To deal with such structural differences, data integration could be achieved by bringing heterogeneous models in conformance to an upper model. For example, the Structured Metrics Metamodel (SMM) developed at OMG for representing measurement information has a DimensionalMeasure class that is a specialization of the Measure class and has an attribute unit. Then, data stored through the relation Unit in FuGE-OM (Figure 2.a) and class Unit in MAGE-OM (Figure 2b) could be bridged through SMM.

At this point, model-based approach of structural heterogeneity might be viewed just as increasing the abstraction level whose side effect might be to reduce complexity; but it is without counting on semantic heterogeneity. In many cases, not only the differential use of building blocks is problematic but the meaning of words used for naming concepts may differ across communities.

Semantic heterogeneity clearly stands out as the most important and more crucial issue and a strong obstacle to circumvent. However, to return the strength of evidence, semantics is context-dependent and the meaning of concepts/models often requires a deep understanding of roles and relations to other concepts/models in a specific domain. In other words, semantics is context-sensitive, which is fully sufficient to justify designing, on a case-by-case basis, domain-specific metamodels. Thus, in Health sciences, which is our domain of interest, several metamodels will track different perspectives including patient profiles, biological features, clinics, etc., all based on metadata/data standards and domain ontologies. An example of this approach is given in [2] and the proposed architecture was organized as follows:

- The upper general domain-specific metamodel was the FuGE framework [23] denoted mm_FuGE; it was developed to specify high-throughput data production about genome-wide biological components;
- Sub-families that recognize FuGE’s extension guidelines (moreover, they correspond to sub-domains of expertise) were sharing more precise consensus with mm_FuGE; they were denoted “reusable models” (rm); MAGE specification [24] constitutes such a sub-family and it was denoted rm_MAGE;
- Models were derived from reusable models: GEO [25] and ArrayExpress [26] applications were developed according to the MAGE specifications. Five other applications were designed in line with the PSI/MI [27] specifications that define another sub-domain of expertise in high-throughput biology.

Thus, the above metamodeling architecture facilitates semantic integration by precisely defining what was shared and what was not shared by two applications. For example, the reusable model rm-PSI/MI was shared by the IntAct and MIPS applications, while the rm_FuGE-extension reusable model was the only model shared by the ArrayExpress and GEO applications.

In another work, we have used two frameworks specifying the same domain of discourse to operate semantic integration of data files to any format: the FuGE frame described in the object-oriented format and the ISA-TAB frame [3] described in the tabular format were used as metamodels to elicit model transformation (specification document for an ISA-TAB2FuGE transformation is available at [http://www.biodataconsulting.com](http://www.biodataconsulting.com)). Data files were made accessible under both formats [3].

In spite of efforts in standardization, maintaining the semantic quality of integrated models is hard to assure. In order to achieve this semantic quality, general ontologies as the Basic Formal Ontology (BFO) [28] might be helpful as a semantic guideline.

More generally, alignment with general ontologies is likely to provide good stability under diverging evolutions of biology sub-domains. Since biological data tend to be complex, major building blocks (e.g., representations of molecular sequences) generally depict several views which can differ from one model to another. Alignment of such views on concepts forming the core of a general ontology helps to guarantee their stability.
4 Conclusions

4.1 Three reasons to ensure continuing

We are strongly convinced that MDE concepts, methods and tools will help adding yet another approach to data integration. First and foremost, it makes of needs and solutions specifications an integral part of the developing process (and not only for reporting), that is fundamental because we are running in difficulties with the increase in data volume and their semantic complexity; in addition, cross-disciplinary expertise are required to address these complex issues and various skilled groups must clearly understand common challenges and opportunities.

Second, even if most of the technical problems in data integration have now been overcome, the same is not true for semantic aspects. Current approaches are built on mediator-based data integration system that may use ontologies as common schemas or multiple ontologies above each data source, etc. As metamodels are simplified ontologies, we think that model-driven approaches are suited for semantic data integration by its very nature.

Last and not least, model-driven approaches are incremental; model architectures are set up in a modular manner and models are made, notably, by (sub)model aggregation. This Lego-like approach is well in-line with the way knowledge is generated and acquired, especially in complex domains such as Health.

4.2 Future work

Our motivation is to use model-driven data integration in various domains, notably Life Sciences with special attention to Health.

For example, data will be collected from remote sources on people and volunteers (civil registrar, address, environment (industrial and domestic waste, urban pollution,)), health (disease, follow-up), geolocation (place points from radiofrequency identification tag,...), etc., for further analyses in various contexts.

More precisely, our approach will be used to integrate macroscopic (clinical and physiological), molecular (biological), environmental (managed by geomatics domain) data on patients to address knowledge in Health and Medicine through systemic approaches. The main challenge is to better understand diseases while discovering new assets for prevention and therapy.

5 References


