Towards a Framework for Federated Interoperability to Implement an Automated Model Transformation

Mustapha Labreche, Aurelie Montarnal, Sébastien Truptil, Xavier Lorca, Sébastien Weill, Jean-Pièrre Adi

Towards a Framework for Federated Interoperability to Implement an Automated Model Transformation

Mustapha Labreche[1](✉), Aurélie Montarnal[1], Sébastien Truptil[1], Xavier Lorca[1], Sébastien Weill[2], and Jean-Pièrre Adi[2]

[1] IMT Mines Albi, 20 Chemin de La Teulière, 81000 Albi, France
{Mustapha.Labreche,Aurelie.Montarnal, Xavier.Lorca}@mines-albi.fr, Sebastien.Truptil@cea.fr
[2] Sylob, 7 Rue Marcel Dassault, 81990 Cambon, France
Sebastien.Weill@forterro.com, jpa@sylob.com

Abstract. In order to adapt to changes in industrial world (customers and markets) and to competition, to create economic or strategic partnerships with external players or simply to integrate a connector to exchange information between the various services and software of a company, it is essential to have the necessary software tools (by development or deployment) that guarantee effective communication between the various parties, which are often heterogeneous and not known in advance, and overcome certain difficulties such as the multiplicity of information sources and the quality of the data.

In such a context, the exchange or migration of data is a critical step. In order to facilitate the exchange, our approach aims at implementing federated interoperability with automated model transformation, supported by an interoperability evaluation, in order to ensure data retention despite the unforeseen uses (for example, some fields divert or mislabel) and to ensure their consistency during the transformation.

Keywords: Model-driven engineering · Automatic transformation · Federated interoperability · Interoperability evaluation

1 Introduction

Collaborative networks (CN) represent a set of heterogeneous organizations with different services, skills and systems, but who share a common interest [1]. In this context, companies are increasingly turning towards collaborations with an opportunistic dynamic of creation and dismantling in order to benefit from new know-how and to propose new offers [2]. For this, in order to communicate between partners in a CN, it is necessary to be able to quickly and efficiently collaborate. So, a way to translate data on the fly is needed to ensure smooth collaborative workflows.

This observation has repercussions on information systems and more particularly on data exchange. Indeed, information systems must be able to exchange and use information regardless of the data format without requiring laborious human actions.

https://doi.org/10.1007/978-3-030-62412-5_12

Thus, the issue of interoperability of systems is defined by [3] as «The ability of two or more systems and components to exchange and reuse the information».

However, the implementation of interoperability is not trivial because of the presence of barriers: conceptual barriers (semantic and syntactic incompatibility of information), technological barriers (incompatibility of platforms or architectures), and organizational barriers (incompatibility of structures and management techniques) [4]. Establishing interoperability is therefore about removing barriers, and to this end there are three fundamental approaches to achieving interoperability [4]:

- Integrated approach: the different parties use a common format for the information accepted by all parties.
- Unified approach: the parties define a common format and must then translate their data into the common template before exchanging it.
- Federated approach: in this approach there is no common format. To establish interoperability, the parties must adapt on-the-fly and no party imposes any model or working method and must translate the data received.

The paper proposes an approach to facilitate the exchange or transfer and integration of data from one system to another with different data formats. Therefore, trying a manual solution is error-prone and time-consuming, and is difficult to reproduce in another context (due to the specific rules imposed). The contribution of the paper is a methodology for implementing federated interoperability "on-the-fly" through an automated model transformation, supported by a three-level interoperability assessment model, which will have to answer the following questions:

1. How to ensure federated interoperability?
   - How to manage the heterogeneity of systems?
   - How to ensure the coherence of transformed data?
2. What are the qualitative or quantitative evaluation criteria for interoperability?

2 Related Works and Technical Recall

Interoperability is a prerequisite to reduce the difficulty of collaboration. The approach presented in the paper deals with federated interoperability with an emphasis on the heterogeneity of information systems. This section will provide an overview of research activities in this field and will outline some models for interoperability assessment, frameworks for developing interoperability, and some methodologies to solving complex interoperability problems [5]:

2.1 Interoperability Assessment

In order to be able to improve interoperability, it is important to be able to evaluate it. For a company, the evaluation will allow it to know its strengths and weaknesses for a partnership with other companies [5]. Interoperability can be measured qualitatively or quantitatively and assessed on three levels [6]: (1) the maturity assessment which aims to evaluate the company's capacity to adapt to possible partnerships, (2) the compatibility assessment which allows to calculate the degree of compatibility between two systems and (3) the performance assessment which evaluates the performances during the partnership.

There are several approaches in the literature, including Level of Information System Interoperability (LISI) [7] which is one of the first maturity models for assessing interoperability between different information systems (Organizational Interoperability Maturity Model (OIMM) extends this approach by including organizational interoperability [8]). Other approaches consider only the assessment of semantic interoperability [9] or conceptual interoperability [10]. While others cover the three interoperability barriers [11] or the three levels of assessment [12].

2.2 Interoperability Frameworks

Interoperability Development for Enterprise Application and Software (IDEAS) [12] is a Framework that proposes to establish interoperability based on three layers: Business, Knowledge and Information and Communications Technology (ICT) by considering the semantic dimension on each of the layers and takes into account other qualitative attributes (security, performance, portability …) in order to develop a common understanding.

European Interoperability Framework (EIF) [13] proposes a framework that identifies a set of standards and recommendations that guide the way European public services cooperate online by defining three types of interoperability: organizational, semantic and technical.

INTEROP-NoE [14] defines a framework for delivering interoperability solutions within an enterprise based on two dimensions: interoperability barriers and interoperability levels. The barriers (organizational, conceptual and technological) relate to the problems that block the development of interoperability and the levels (business, processes, services and data) represent the different views where interoperability can be implemented. This framework has become a standard (ISO 11354-1) for which several works on interoperability have been based and presented in the literature [15].

2.3 Interoperability Methodology

A methodology is built to analyze and design a system to solve complex problems [5]. Developing a methodology for enterprise interoperability involves efficiently designing, structuring, aligning and implementing a set of methods, models and approaches from several scientific fields [16] to overcome the different barriers and achieve interoperability at each level. In that way, model-driven (MD) approaches are frequently used.

Model-Driven Engineering (MDE) is a high-level abstraction approach that applies the benefits of modeling to software engineering activities and where the notion of *model* is a central concept for which there is no universal definition [17]. In [18] «A Model is a description of a system, where 'system' may include not only software and hardware but organizations and processes. A model is represented by a set of elements that are structurally defined by properties and interrelated by relationships. Thus, the set of model elements conforms to the complete representation of the modeled concerns, a model is said to conform to a *metamodel* that defines both aspects of the model, namely syntax and semantics [19].

A metamodel is therefore an explicit model of the elements and rules needed to build specific models, but like the definition of a model, there are a variety of definitions of metamodel. We can then say that if a metamodel is a model of a *modeling language*, there must be a metamodel describing the metamodel and so on [20].

Several concepts based on MDE have emerged:

– Model-Driven Reverse Engineering (MDRE): allows the obtaining and reuse of representations (e.g. source code or configuration file from legacy system).
– Model-Driven Interoperability (MDI): it aims at defining solutions to achieve interoperability between two or more systems by applying MD techniques.

These concepts use model transformation which aims to make models usable which consists in ensuring the passage from one model to another (from a source model to a target model) sharing the same concepts and using transformation rules implemented by a transformation language [21].

In general, model transformation approaches fall into three categories [22]: (1) model-to-text (M2T) often used in MDRE to transform one or more source models to generate scripts, configuration or documentation files, (2) text-to-model (T2M) which takes a text format as input and transforms it into one or more models, (3) model-to-model (M2M) which transforms one or more source models (conforming to a source metamodel) to one or more target models (conforming to a target metamodel).

Various methodologies using these approaches and their concepts are proposed:

HLA Federated Interoperability Framework [23] is one of the most popular methodology for designing complex systems. It is a methodology for the interoperability of heterogeneous information systems and it synchronizes a high-level architecture with the model-driven approach extended to the Web service.

[24] Interoperability Systems Integration and Re-Engineering (IRIS) is a structured approach that defines the tools to be used in each phase of an enterprise interoperability project taking into account different interoperability views, such as Business, Process Management, Knowledge, Human Resources, ICT and Semantics.

Although each methodology has its advantages and provides good theoretical insights, most of them seem to lack concrete application cases. Moreover, to our knowledge, it appears that no methodology or framework is able to address the problem of federated interoperability on the fly between heterogenous systems. Therefore, the use of MD approaches and graph theory are promising way to address this type of problem. Indeed, model-driven approaches allow to formalize and manipulate different levels of abstraction within the framework of federated interoperability through the use of meta-models and their instances in the form of models, and graph theory allows to represent heterogenous systems on the fly and to provide algorithms for evaluating interoperability. In the following section, this combination approach is discussed in detail.

3 Towards an Approach of Representation and Exploitation by Graphs

The paper presents a global approach for establishing and evaluating federated interoperability between heterogeneous systems using graph theory. Indeed, reasoning on graphs to make transformations, detect conflicts, model relationships, find or compute optimized indicators is interesting since they are on the one hand a mathematical modeling tool presenting a visual aspect of meta-models and their related instances for analysis and on the other hand a tool for evaluating the different connections between data, models, meta-models and heterogeneous systems.

Modeling in the form of graphs is obtained by transforming the elements of a model into a set of nodes and the various links between the elements by edges. Then, the transformation from one graph to another representing the source and target models is done by applying a sequence of conversion and mapping rules based on semantics. Thus, the combination of model transformation and graph transformation technics is beneficial since the former is very practical and the latter is very expressive. The details of the approach will be described in the next sections along 3 stages: (i) the database exploration allows discovering the source and target data models (frequently needed as databases tend to evolve overtime and gradually lose their coherency); then, (ii) the metamodel and model representation as graph can be achieved; finally, (iii) a matching analysis helps define the rules that will be used for the transformation of a source database to the target expected database structure.

3.1 Database Exploration and Model Analysis

This step starts with moving from the database to the models, then in discovering and understanding the model constructed by analyzing the structure in order to identify the components of the system and/or to construct the relationships between these components by creating a representation at a higher level of abstraction.

As an example, the case of two relational databases extracted from the system but deployed in two different software versions. In the case of another type of data, the process remains unchanged, only the metamodels to which the models conform are modified.

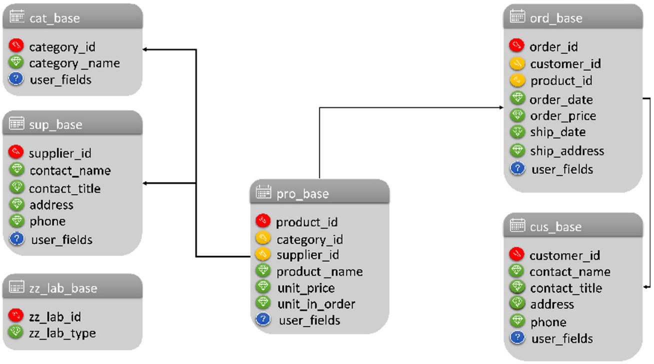

Fig. 1. An example of relational database represented by database scheme (Color figure online)

3.1.1 Database Representation

In a relational database the information is organized in the form of tables (also called relations) where each table is composed of one or more rows representing the records (tuples) and one or more columns listing the attributes that relate to a record.

A key is an attribute (or group of attributes) that uniquely identifies a row in a table. A key is said to be primary and is unique if one of the keys specifies the sequence of data records in the table.

Finally, the relationship between tables can be defined using foreign keys, which is simply a primary key of another table and which also allows correlation between different tables in a database.

An example of such database schema is presented in the Fig. 1, where each table has a set of attributes, a primary key (PK) (red icons) and foreign key(s) (FK) (yellow icons). Some tables have attributes denoted “user_fields” (blue icons) and other tables are prefixed with “zz_…. “ which are specific attributes and tables created for customizable use, i.e., created for a specific need. However, they are sometimes diverted from their default function.

3.1.2 Model Analyses

Once the models are built, the next step would be to explore them. In our study this consists of analyzing the input and output models in order to identify and establish relationships at two levels: at the data model level and then at the data level. The linking can be guided by the construction of a weaving model [25] that allows capturing different types of links (semantics, composition, interoperability, data integration, traceability and ontological alignment) [26].

Another step is carried out in order to manage the various conflicts generated by the different information representations between model elements (hierarchization of the same information, aggregation, data types, etc.) [27]. Also, some links can be created and added manually by experts in the field.

At the end of this step, the constructed links allow a better understanding of the heterogeneous parts by exploiting the information specific to each system and between systems. This analysis can be translated into configurations that induce semi-automation/automation of the transformation process.

3.2 Model/Metamodel Modeling

In order to give the approach leverage to adapt to unknown models/metamodels in advance, it is essential to use a solution that allows on-the-fly model transformations to be generated. In other words, the approach must be able to represent different information and apply algorithms in a generic way. This dynamic feature can be offered using graph theory.

To modeling metamodels/models, the class of attributed type graphs is used since it offers modeling formalisms for the different structures (attribute, relation, class, table).

This graph class is characterized by two parts:

1. a part which describes the graph: composed of nodes (graph node) representing tables and edges (graph edge) representing the links between these tables (for example: inheritance or primary keys) and
2. a data description part: composed of nodes (data nodes) representing the data types (string, date…) and edges (node attribute edge) linking the nodes of the graph to the data nodes.

Figure 2 describes simple example of transforming three tables into a graph (Fig. 3). The tables are characterized by and keys (primary and foreign), attributes each having a type (string, date, int,…). The elements of the graph are described in Fig. 4.

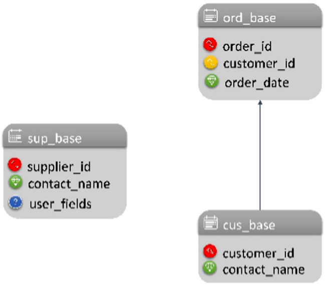

Fig. 2. A simple example of relational database

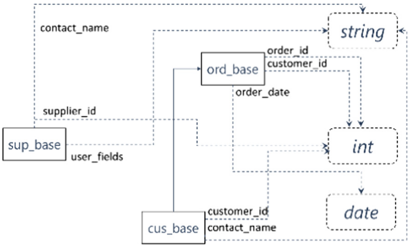

Fig. 3. Graph representing the database

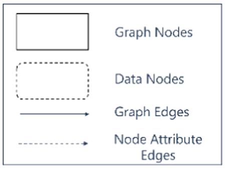

Fig. 4. The description of the different elements of the graph

3.3 Matching and Transformation

Matching is a process that aims to identify and discover correspondences between two or more agents (documents, images, systems…). Matching algorithms are used in several domains and are based on several principles (syntactic, semantic, numerical), which is why various solutions and implementation policies are proposed [28]. The matching ensures a connection (if established) between the agents by means of transformation rules. Thus, matching ensures the interoperability of information systems.

In order to build transformation rules, it is necessary to create a mapping (configuration) between the elements of the metamodels. To do so, we must compute the similarity between the elements (nodes and edges) of the two corresponding graphs using semantic-based matching techniques and algorithms that perform on the graphs.

Once the mapping is done, the final transformation rules are created and defined with the help of the preliminary analysis performed beforehand and some detection measures predefined by experts in the field.

3.4 Interoperability Assessment

From a collaborative perspective, it is important to measure the capacity of collaboration between enterprises by assessing continuously interoperability of (heterogeneous) systems, architectures and services [14] and detect potential problems.

Assessing interoperability will be on three levels: for the first level the objective is to calculate the ability of systems to communicate or exchange and to detect potential problems (can we interoperate or not? If not, what can we do?). In the second level, the measurement will be done continuously on indicators probably defined and fixed in order to detect potential blockages and barriers and try to overcome them (if possible). Finally, in the third level, the objective is to measure the final interoperability (has optimal performance been achieved?)

An approach to assessing interoperability is the assessment of semantic interoperability. The matching approach allows the integration of various measures of similarity and dissimilarity between graphs. Indeed, since we aim at establishing the different semantic correspondences between models, it is quite possible to have redundant, erroneous or incomprehensible information problems which are solved by ontology-based approaches [5]. Calculations using functions based on linguistic and structural similarity [15] and the accuracy or relevance (business aspect) of the information are then proposed as quantitative and qualitative evaluation criteria, that can be in turn used as performance and validation criteria of the proposed methodology.

4 Conclusion and Future Work

The aim of this work is to show that the combined adoption of concepts from graph theory, Model-driven engineering and Model-driven reverse engineering allows both facilitating and establishing on-the-fly federated interoperability and makes the implementation of evaluation mechanisms easier and more uniform between heterogeneous systems.

Such an approach is still in progress and is in development within the framework of a real case study of database transformation relating to the same system, but deployed in two different software versions, with data models that are also different and more or less “customized” by customers. The next steps of the research will thus focus on the implementation of the proposed methodology and its validation thanks to the interoperability assessment criteria.

Finally, this framework can be adapted and applied in the context of a software upgrade, technical migration or implementation of connectors to ensure communication with third-party services.

Acknowledgement. This work is funded by the company Forterro Sylob, which will bring in real case studies.

References

1. Camarinha-Matos, L.M., Fornasiero, R., Afsarmanesh, H.: Collaborative networks as a core enabler of Industry 4.0. In: Camarinha-Matos, L.M., Afsarmanesh, H., Fornasiero, R. (eds.) PRO-VE 2017. IAICT, vol. 506, pp. 3–17. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65151-4_1
2. Dodgson, M.: Technological Collaboration in Industry: Strategy, Policy and Internationalization in Innovation, vol. 11. Routledge (2018)
3. Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. IEEE (1990)
4. Chen, D., Doumeingts, G., Vernadat, F.: Architectures for enterprise integration and interoperability: past, present and future. Comput. Ind. **59**(7), 647–659 (2008)
5. Fortineau, V., Paviot, T., Lamouri, S.: Improving the interoperability of industrial information systems with description logic-based models—the state of the art. Comput. Ind. **64**(4), 363–375 (2013)
6. Leal, G.D.S.S., Guédria, W., Panetto, H.: Interoperability assessment: a systematic literature review. Comput. Ind. **106**, 111–132 (2019)
7. DoD, U.S: C4ISR Architecture Working Group (AWG): C4ISR Architecture Framework Version 2.0, Washington DC, December 1997
8. Fewell, S., Clark, T.: Organisational Interoperability: Evaluation and Further Development of the OIM Model. Defence Science and Technology Organisation Edinburgh, Australia (2003)
9. Dibowski, H.: Semantic interoperability evaluation model for devices in automation systems. In: 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1–6. IEEE, September 2017
10. Tolk, A., Muguira, J.A.: The levels of conceptual interoperability model. In: Proceedings of the 2003 Fall Simulation Interoperability Workshop, vol. 7, pp. 1–11. Citeseer (2003)
11. Guédria, W., Naudet, Y., Chen, D.: Maturity model for enterprise interoperability. Enterp. Inf. Syst. **9**(1), 1–28 (2015)
12. IDEAS Consortium: IDEAS Project Deliverables (WP1-WP7). Public Reports (2003)
13. IDABC, Enterprise, and D. G. Industry: European interoperability Framework for pan-European e-government services. European Communities (2004)
14. Panetto, H., Scannapieco, M., Zelm, M.: INTEROP NoE: interoperability research for networked enterprises applications and software. In: Meersman, R., Tari, Z., Corsaro, A. (eds.) OTM 2004. LNCS, vol. 3292, pp. 866–882. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30470-8_100
15. Neghab, A.P., Etienne, A., Kleiner, M., Roucoules, L.: Performance evaluation of collaboration in the design process: using interoperability measurement. Comput. Ind. **72**, 14–26 (2015)
16. Ensemble, C.: EISB Basic Elements Report. Envisioning, supporting and promoting future internet enterprise systems research through scientific collaboration (FP7-ICT-257548), deliverable D, vol. 2 (2011)
17. Da Silva, A.R.: Model-driven engineering: a survey supported by the unified conceptual model. Comput. Lang. Syst. Struct. **43**, 139–155 (2015)
18. UML, OMG: Unified Modeling Language (OMG UML). Version 2.5. 1 UML (2017)

19. Kleppe, A.: A language description is more than a metamodel. In: Fourth international workshop on software language engineering, vol. 1, megaplanet.org, October 2007
20. OMG: Meta object facility specification. OMG document formal, version 1.4 (2003)
21. Czarnecki, K., Helsen, S.: Feature-based survey of model transformation approaches. IBM Syst. J. **45**(3), 621–645 (2006)
22. Kahani, N., Bagherzadeh, M., Cordy, J.R., Dingel, J., Varró, D.: Survey and classification of model transformation tools. Softw. Syst. Model. **18**(4), 2361–2397 (2018). https://doi.org/10.1007/s10270-018-0665-6
23. Tu, Z., Zacharewicz, G., Chen, D.: Building a high-level architecture federated interoperable framework from legacy information systems. Int. J. Comput. Integr. Manuf. **27**(4), 313–332 (2014)
24. Chalmeta, R., Pazos, V.: A step-by-step methodology for enterprise interoperability projects. Enterp. Inf. Syst. **9**(4), 436–464 (2015)
25. Del Fabro, M.D., Bézivin, J., Valduriez, P.: Weaving models with the eclipse AMW plugin. In: Eclipse Modeling Symposium, Eclipse Summit Europe, vol. 2006, pp. 37–44, October 2006
26. Yie, A.: A non-invasive approach for evolving model transformation chains. Doctoral dissertation, Universidad de Los Andes (2011)
27. Kataria, P.: Resolving semantic conflicts through ontological layering. Doctoral dissertation, University of Westminster (2011)
28. Emmert-Streib, F., Dehmer, M., Shi, Y.: Fifty years of graph matching, network alignment and network comparison. Inf. Sci. **346**, 180–197 (2016)