Towards an MDE-Based Approach to Test Entity Reconciliation Applications

J.G. Enríquez[1], Raquel Blanco[2], F.J. Domínguez-Mayo[1], Javier Tuya[2], M.J. Escalona[1]
[1]Department of Computer and Language Systems, University of Seville, Seville, Spain
[2]Department of Computing, University of Oviedo, Gijón, Spain
jose.gonzalez@iwt2.org, {rblanco, tuya}@uniovi.es, {fjdominguez, mjescalona}@us.es

ABSTRACT
The management of large volumes of data has given rise to significant challenges to the entity reconciliation problem (which refers to combining data from different sources for a unified vision) due to the fact that the data are becoming more unstructured, unclean and incomplete, need to be more linked, etc. Testing the applications that implement the entity reconciliation problem is crucial to ensure both the correctness of the reconciliation process and the quality of the reconciled data. In this paper, we present a first approach, based on MDE, which allows the creation of test models for the integration testing of entity reconciliation applications.

Keywords
Entity Reconciliation, Software Testing, Big Data, MDE.

1. INTRODUCTION
Currently, information management is critical in many aspects of our lives. However, the incorporation of information and communications technology (ICT) into everyday life causes people to experience an overload of information, also known by the term "infoxication". This term refers to the difficulty that someone has in understanding a problem and making decisions about it because of an excess of information [25].

In the first era of ICT, the main problem that researchers had was how to find information and how to store and manage it efficiently. Currently, due to the presence of Big Data and cloud computing, the biggest problem is how to extract knowledge of the information based on the needs of each user [6]. In this sense, the problem of reconciling entities takes on a very important role.

Entity reconciliation (also called entity resolution or ER) is a fundamental problem in data integration. It refers to combining data from different sources for a unified vision or, in other words, identifying entities from the digital world that refer to the same real-world entity. It is an uncertain process because the decision to allocate a set of records with the same entity, cannot be taken with certainty, unless these records are identical in all their attributes or they have a common key [10][21]. This problem can be applied to many kinds of scenarios. An example of entity reconciliation is given in Figure 1. At left of the figure, there are two different data sources with information related to the names of the authors of this paper. In each data source, the signatures are different, but they are related to the same authors. Due to the reconciliation of the entities stored in each data source, we can obtain a simpler model where the information of each entity is stored in just one database.

While this problem is not new, the management of large volumes of data presents new challenges and the necessity of carrying out a high quality reconciliation of entities is growing in the era of Big Data [6][8]. In [11], the authors expose some of the main challenges of entity reconciliation in the Big Data environment such as: *data heterogeneity*, it is becoming more common that data are unstructured, unclean or incomplete and also there are diverse data types; *data more linked*, where it is expressed the necessity of inferring relationships; *multi-relational data*, dealing with the structure of entities; and *building multi-domain systems*, trying to customize methods that span across domains. In the literature it is possible to find a wide variety of approaches to try to solve the problem of reconciliation of entities, such as: deterministic rule-based methods [14][9][2], probabilistic-based methods [24][20][7], and learning-based [17][5] and graph-based methods[13][21][22].

Due to the important challenges of the ER problem, it is crucial to test the operations designed to carry out the reconciliations and the applications that implement them in order to ensure both the correctness of the reconciliation and the high quality of the reconciled data.

In this paper, we propose an approach based on the Model-Driven Engineering (MDE) paradigm for testing applications that implement ER problems. The approach relies on the ER problem specification and the conceptual data models of the sources and the solution to be achieved in order to define test models composed of a set of business rules, which specify the system requirements. From these business rules, the situations of interest to be tested (*test requirements*) that guide the generation of the test cases can be automatically derived.

MDE [18] emerged to address the complexity of software systems and to express the concepts of the problem domain in an effective way. In this vein, the basic principle of MDE is "Everything is a model" [1]. The main idea of the MDE is to use a set of models to decrease the level of abstraction. Thus, in the early stages of development, models are more abstract than in the final stages where the models are much closer to implementation. One of the advantages of MDE is its support for automation, as the models can be automatically transformed from the early stages of development to the final stages. Therefore, MDE allows automating the tasks involved in a software development, such as the testing tasks.

The main contributions of this work are:

- The definition of a framework that includes the integration testing process into the entity reconciliation process.

- The definition of a test model that represents the testing objectives as business rules, which can be used to automatically derive the test requirements.

The remainder of this paper is organized as follow: Section 2 presents the problem approach. Section 3 describes the testing metamodel for the entity reconciliation, which is still in-progress. The paper ends with conclusions and a summary of future works.

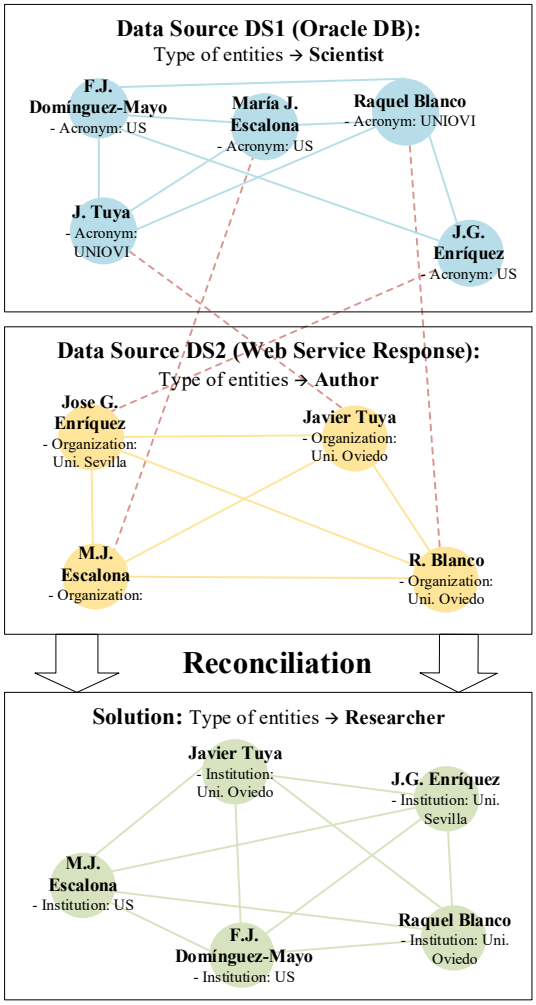

Figure 1. Entity reconciliation example

2. PROBLEM APPROACH

In a previous work [6], we proposed an approach to address the problem of reconciliation of entities based on MDE and virtual graphs technology. The proposal presented in this paper extends the previous work by adding a new fundamental pillar in the reconciliation of entities: testing. We aim to ensure the quality of the entity reconciliation process that is developed.

Graph technology is a natural solution to addressing problems related to Big Data and especially for the relationships between entities. The wide variety of existing algorithms, for example: Dijkstra, A*, Kruskal, etc. offer great flexibility in different situations. Theoretically, graphs can be displayed in two ways: explicit and implicit. An explicit graph is a collection of items (vertexes and edges) that can be stored in memory, which means that each vertex and each edge of the graph can be completely stored in memory. An implicit (or virtual) graph is a graph that cannot be completely stored in memory for various reasons, such as size or hardware limitations [15].

Thus, a virtual graph is the data structure defined for representing the information that forms the solution of the entity reconciliation process. With this technology, we have the possibility of building the structure on the fly. This lets us build different solutions to address many scenarios within a business logic where the predefined data model cannot meet the extensibility or availability of the required data sources.

Figure 2 depicts the architecture of our proposal, which allows the user to create models to address the entity reconciliation and to lead the testing of this process. The four pillars of this architecture are the following metamodels:

- *Virtual Graph metamodel*: allows the user to design the conceptual data model that represents the solution to be achieved, according to the ER problem domain. This metamodel is an extended version of a graph metamodel.
- *Data Sources metamodel*: allows representing the information of the data sources to be reconciled as well as the way of accessing them. These sources can be a structured or unstructured database, a web service, a warehouse or another information generator.
- *Transformations metamodel*: represents the different transformations that the data of the sources must undergo in order to carry out the entity reconciliation and to be consistent with the data model of the solution (represented by an instantiation of the virtual graph metamodel).
- *Testing metamodel*: allows representing the testing objectives for the entity reconciliation, such that it can be determined in the early stages of the development if such reconciliation is what the user really wants to carry out and if the results obtained are the expected ones.

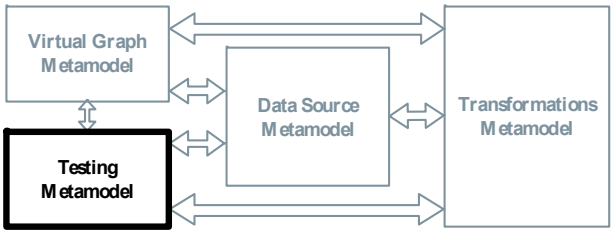

Figure 2. Architecture of the proposal

3. TESTING METAMODEL FOR THE ENTITY RECONCILIATION

Our approach allows the creation of test models for integration testing, which represent the testing objectives from the entity reconciliation specification and the conceptual data models of both the data sources to be reconciled and the solution to be achieved. These test models are composed of several business rules. The business rules are statements that define or constrain the business structure or the business behaviour [12], and have been used in other approaches focused on testing database applications [3], [23].

The business rules of our approach, called *integration rules*, impose conditions on: (1) the structure of the solution, (2) the data that address the reconciliation process, (3) the data that constitute the solution, and (4) the business logic of the reconciliation process. The elements of an integration rule are depicted in the metamodel of Figure 3.

The data sources to be reconciled and the solution to be achieved can have different types of entities and several entities of each of them. For example, in the relational data models, the different types

of entities correspond to different tables and the entities of a specific type correspond to the tuples of a table. In the graph data models, the different types of entities correspond to different types of nodes and an entity corresponds to a specific node. When an integration rule is defined, it is necessary to specify the set of entities and relationships that are affected by the conditions that the rule imposes, which constitute its reconciliation scope. This scope is called *integration context* and it is represented by the metaclass *IntegrationContext*.

In our work-in-progress we distinguish two main types of integration rules:

- *Structural rules* (represented by the metaclass *Structural*) constrain the structure of the solution to be achieved and impose conditions to identify the entities and relationships of the data sources that derive the new entities and relationships to be included in the solution.
- *Load rules (represented by the metaclass Load)* establish conditions to be fulfilled by the attributes of the entities that constitute the solution, regarding the values of some specific attributes of the data sources involved in the reconciliation process. They also can impose pre-conditions on the attributes of the data source that must be fulfilled to load new data into the solution.

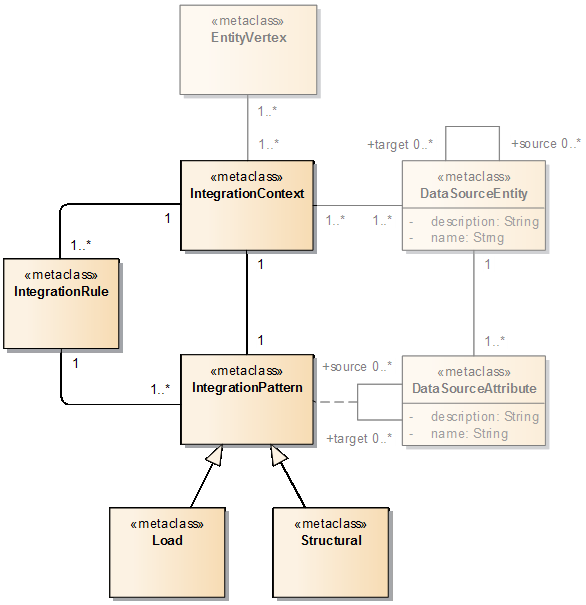

Figure 3. Testing metamodel

We are working on the definition of several types of structural rules, considering the structural elements of the solution (entities and relationships), and also on several types of load rules, taking into account different patterns that can be used to constrain the values of the attributes in the solutions and the pre-conditions.

After defining the integration rules that constitute the test model, test selection criteria can be applied over the conditions imposed by the integration rules to derive the situations of interest to be tested (the test requirements). Then, these test requirements are used to guide the generation of the test cases. To automate these processes, transformations guided by some test selection criterion and transformations guided by a test generation technique must be defined, respectively.

Consider Figure 1 to illustrate the process of defining the integration rules and deriving the test requirements and the test cases. The structural rules impose conditions to unify the entities according to a specific degree of similarity (for example, the entities "J.G. Enríquez" and "Jose G. Enríquez" are unified into the entity "J.G. Enríquez"). They also establish conditions to create the relationships between the entities of the solution, taking into account the relationships of the data sources that relate the entities that have been reconciled (for example, the relationship between "J.G. Enríquez" and "Javier Tuya").

An example of a load rule written in a language based on the SBVR specification [16] is depicted in Figure 4. Its goal is to specify the value of the attribute "Institution" of each entity "Researcher" included in the solution. Statements 1 to 3 specify the integration context, which is formed by the paths P1 and P2. These paths determine the entities of the data sources DS1 and DS2 that have been reconciled into the entities of the solution (see Table 1). Statements 4 and 5 specify the prioritization of the attributes "Organization" and "Acronym" and statement 6 establishes that the attribute "Institution" of each entity of the solution can only obtain its value from one of these source attributes.

(1) Path P1 is Solution.Researcher [$f_{similarity}$(Researcher.name, Scientist.name)] DS1.Scientist
(2) Path P2 is Solution.Researcher[$f_{similarity}$(Researcher.name, Author.name)]DS2.Author
(3) Integration context IC is P1, P2
(4) IC.organization has priority 1
(5) IC.acronym has priority 2
(6) Each IC.institution is only IC.organization or IC.acronym

Figure 4. Example of a load rule

Table 1. Integration context of the load rule

	Solution	DS1		DS2	
	Name	Name	Acronym	Name	Organization
1	Raquel Blanco	Raquel Blanco	UNIOVI	R. Blanco	Uni. Oviedo
2	M.J. Escalona	Maria J. Escalona	US	M.J. Escalona	
3	F.J. Domín-guez-Mayo	F.J. Domín-guez-Mayo	US		
4	J.G. Enríquez	J.G. Enríquez	US	Jose G. Enríquez	Uni. Sevilla
5	Javier Tuya	J. Tuya	UNIOVI	Javier Tuya	Uni. Oviedo

To derive the test requirements from an integration rule, we apply the MCDC criterion [4] over the conditions imposed by the integration context and the structural/load rule. This coverage criterion has demonstrated its utility in previous work, such as [19] (for testing SQL queries) and [3] (for testing the user-database interaction). To automatically apply this criterion and check the test coverage, both integration rules and test requirements can be transformed into an executable representation [3]. For the example of Figure 4, some test requirements are:

(1) Both "Organization" and "Acronym" have a value in the source entities reconciled in an entity of the solution.
(2) "Organization" does not have a value and "Acronym" has a value in the source entities reconciled in an entity of the solution.
(3) An entity of the solution has been obtained only from an entity of DS1 and "Acronym" has a value.

The test case that covers the previous test requirements is composed of a set of entities stored in the virtual graph that represents the solution and a set of entities stored in each data source. The entities shown in the rows 1, 2 and 3 of Table 1 cover the test requirements 1, 2 and 3, respectively. Thus, it is possible to test whether the application correctly implements the prioritization of the attribute

"Organization". For example, a faulty implementation which does not consider that the attribute "Acronym" must be used when "Organization" has a missing value would be detected by the test case, as its outcome would produce the entities of the virtual graph "M.J. Escalona" and "F.J. Domínguez-Mayo" without a value in the attribute "Institution", instead of the value "US".

4. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a work-in-progress that aims to test applications that implement an entity reconciliation problem to ensure the quality of both the applications and the reconciled data. The approach allows the creation of test models for integration testing, taking into account the problem specification and the data models of the data sources and the solution. These test models are composed of several business rules, called integration rules, which can be used to automatically derive the test requirements. Besides, as the integration rules also describe the business logic of the entity reconciliation process, they can be used to partially derive the implementation of the application.

The proposal is based on two main pillars: MDE and virtual graph. The support of automation of the MDE paradigm allows us to build very scalable solutions at a low cost, whilst the virtual graphs allow us to dynamically build the entity reconciliation solution at runtime.

Future work encompasses several avenues. On the one hand, the definition of different types of structural and load rules, and the definition of the transformations that automate the process of deriving the test requirements and the test cases. Furthermore, the extension of the testing metamodel to cover the unit testing of the transformations applied over the data to carry out the entity reconciliation. In addition, the identification of different case studies to validate the approach. At present, we are working on a real case study in collaboration with the Andalucian Institute of Historical Patrimony and Fujitsu Laboratories.

5. ACKNOWLEDGMENTS

This work was supported by projects: PERTEST (TIN2013-46928-C3-1-R), MeGUS (TIN2013-46928-C3-3-R) and SoftPLM network (TIN2015-71938-REDT) funded by the Spanish Ministry of Science and Technology, GRUPIN14-007, funded by the Principality of Asturias (Spain) and ERDF funds.

6. REFERENCES

[1] Bézivin, J. 2005. On the unification power of models. *Software & Systems Modeling*. 4(2), 171-188.

[2] Bhattacharya, I., Getoor, L. 2005. Latent dirichlet allocation model for entity resolution. In *Proc 6th SIAM Int'l Conf. on Data Mining*.

[3] Blanco, R., Tuya, J., Seco, RV. 2012. Test adequacy evaluation for the user-database interaction: a specification-based approach. *In Proc. 5th International Conference on Software Testing, Verification and Validation (ICST 2012)*.

[4] Chilenski, JJ. 2001. *An investigation of three forms of the modified condition decision coverage (MCDC) criterion*. Technical Report DOT/FAA/AR-01/18, U.S. Department of Transportation, Federal Aviation Administration, April 2001.

[5] Cohen, WW., Richman, J. 2002. Learning to match and cluster large high-dimensional data sets for data integration. In *Proc. 8th ACM SIGKDD Int'l conf. on Knowledge discovery and data mining*.

[6] Enríquez, JG., Domínguez-Mayo, FJ., Escalona, MJ., García García, JA., Lee, V., Goto, M. 2015. Entity Identity Reconciliation based Big Data Federation-A MDE approach. In *Proc. Int'l Conf. on Information Syst. Development*.

[7] Fellegi, IP., Sunter AB. 1969. A theory for record linkage. *Journal of American Statistical Assoc*. 64(328),1183–1210.

[8] Gal, A. 2014. Tutorial: Uncertain Entity Resolution. *VLDB Endowment*, 7(13).

[9] Galhardas, H., Florescu, D., Shasha, E., Simon, E., Saita, C. 2001. Declarative data cleaning: language, model and algorithms. In *Proc. Int'l Conf. on Very Large Databases*.

[10] Getoor, L., Machanavajjhala, A. 2012. Entity resolution: theory, practice & open challenges. *VLDB Endowment*. 5(12), 2018-2019.

[11] Getoor, L., Machanavajjhala, A. 2013. Entity resolution for big data. In *Proc. 19th ACM SIGKDD Int'l conf. on Knowledge discovery and data mining.*

[12] Hay, D., Healy, K. 2000. *Defining Business Rules – what are they really?* Technical Report, The Business Rules Group, Revision 1.3, July 2000.

[13] Ioannou, E., Nejdl, W., Niederée, C., & Velegrakis, Y. 2010. On-the-fly entity-aware query processing in the presence of linkage. *VLDB Endowment*, 3(1-2), 429-438.

[14] Lee, H., Chang, A., Peirsman, Y., Chambers, N., Surdeanu, M., Jurafsky, D. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*. 39(4), 885-916.

[15] Mondal, J., Deshpande, A. 2012. Managing large dynamic graphs efficiently. In *Proc. ACM SIGMOD Int'l Conf. on Management of Data*. pp. 145-156.

[16] OMG, Semantics of Business Vocabulary and Business Rules Specification, version 1.3, OMG Document Number: formal/2015-05-0.

[17] Sarawagi, S., Bhamidipaty, A. 2002. Interactive deduplication using active learning. In *Proc. 8th ACM SIGKDD Int'l conf. on Knowledge discovery and data mining*.

[18] Schmidt, D. C. 2006. Guest editor's introduction: Model-driven engineering. *Computer*. 39(2), 25-31.

[19] Tuya, J., Suárez-Cabal, MJ., de la Riva, C. 2010. Full predicate coverage for testing SQL database queries. *Software Testing Verification and Reliability*. 20(3) 237-288.

[20] Verykios, VS., Moustakides, GV., Elfeky, MG. 2003. A Bayesian decision model for cost optimal record matching. *The VLDB Journal*. 12(1), 28-40.

[21] Wang, F., Wang, H., Li, J., Gao, H. 2013. Graph-based reference table construction to facilitate entity matching. *Journal of Systems and Software.* 86(6), 1679-1688.

[22] Wang, H., Li, J., Gao, H. 2016. Efficient entity resolution based on subgraph cohesion. *Knowledge and Information Systems*. 46(2), 285-314.

[23] Willmor, D., Embury, SM. 2006. Testing the implementation of business rules using intensional database tests. In *Proc. Testing: Academic & Industrial Conference on Practice and Research Techniques (TAIC-PART 06)*.

[24] Winkler, WE. 2002. *Methods for record linkage and bayesian networks*. Technical report, Statistical Research Division, US Census Bureau, Washington, DC.

[25] Yang, CC., Chen, H., Hong, K. 2003. Visualization of large category map for internet browsing. *Decis. Support Syst.* 35(1), 89-102.