View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Repositorio Institucional Universidad de Málaga

Analysis of the Scientific Production of the Spanish Software Engineering Community

Loli Burgueño, Antonio Moreno-Delgado, and Antonio Vallecillo

Universidad de Málaga, Atenea Research Group, Málaga, Spain
{loli,amoreno,av}@lcc.uma.es

Abstract. Our group has been working on a report for the Spanish Society of Software Engineering and Software Development Technologies (SISTEDES) to provide a general overview of the Spanish scientific production and its contributions worldwide in the field of Software Engineering. Although a Database solution could have been used, we decided to employ Model-Driven Development (MDD) techniques in order to evaluate their applicability, suitability and fitness for these kinds of purposes, and to learn from the experience in this domain, which combines data integration, large scale models, and complex queries.

Keywords: SISTEDES, scientific contribution, Spain, software engineering, MDD

1 Introduction

The increase in research in the past years and the fact that in most of the cases it is funded by public institutions or private companies raised the need of indicators to measure its quality. Among the most popular are the Journal Citation Report (JCR), which measures the impact factor of a journal, and the *h-index*, which is a measure of the number of high impact papers a scientist has published. Given those metrics it is easy to know whether a publication or a researcher is remarkable. Nevertheless, in the case that the interest resides in knowing the influence of a whole community there is no other option than collecting the information of each member and apply descriptive statistical techniques. Descriptive statistics aims at analyzing a population in order to describe or summarize quantitatively its main features.

Our group has been working on a report for the Spanish Society of Software Engineering and Software Development Technologies (SISTEDES) to provide a general overview of the Spanish scientific production and its contributions worldwide in the field of Software Engineering, using the DBLP database as the source of publications.

Although a Database solution could have been used, we decided to employ MDD techniques to eat our own dog food. We wanted to check whether current MDD technologies and languages could easily deal with these kinds of applications, and to evaluate their applicability, suitability and fitness for these kinds of purposes. This was a good way to learn from the experience in this domain,

which combines data integration, large scale models, and complex queries. This problem also represents several non-trivial challenges. In the first place, the solution needs to be formulated in terms of models and model transformations, where the models integrate data from different heterogeneous sources (DBLP, spreadsheets with information about SISTEDES members, JCR journal rankings). Second, models are too big to be stored in main memory, and to be handled by existing platforms and transformation languages (e.g., QVT [1] or ATL [2]). Thirdly, we need to cope with non-existing or missing data.

In this paper we describe how the above challenges have been addressed, and how the final report has been generated.

The paper is organized as follows. Section 2 presents how the solution to the problem was formulated using MDD techniques, in terms of models that capture the data sources, and transformations implementing the data integration and the queries. Then, Section 3 discusses the queries that were made to the model, and their performan¡ce results. Finally, Section 4 presents our conclusions and an outlook on future work.

2 Analysis definition and procedure

2.1 Modeling the Information Sources

The first step was to identify the information that needed to be handled. Figure 1 depicts the main sources of information and how it is processed.

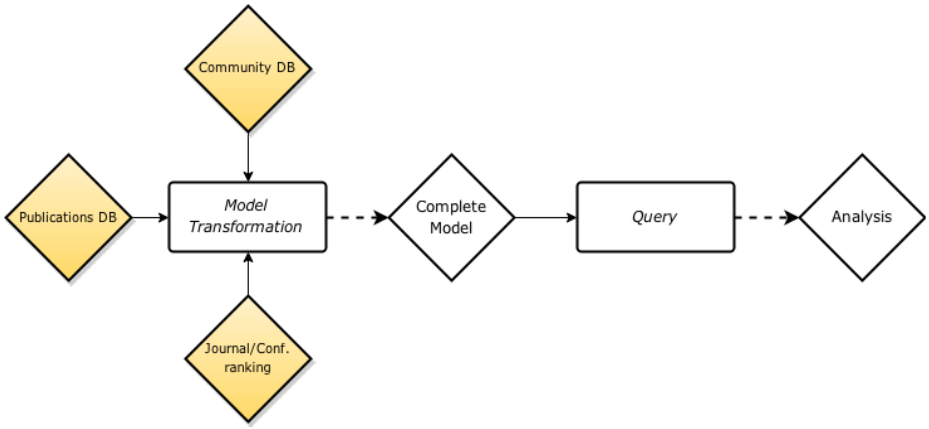


Fig. 1. Information sources and main process.

In the first place, we need the data about the *community* to be analyzed (CommunityDB). In our case study the community consists of the members of

SISTEDES, although our solution does not depend on any particular scientific community, but on the data that we need from their members. This is captured by the Community metamodel shown in Figure 2, which represent Members that belong to Organizations (universities or companies) and can attend the Conferences organized by the community.

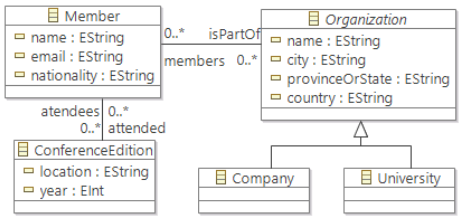


Fig. 2. Community metamodel.

The second source of information is about the publications (Publications DB in Figure 1). In our case it comes from the DBLP database (`http://dblp.uni-trier.de/`). DBLP was selected because it is commonly used in our field, and contains representative information about the contributions that normally matter. Other alternative sources of information that we also considered were Google Scholar, Web of Science or SCOPUS. In any case, in the solution we propose we try to be as agnostic as possible from the concrete publication database, as we shall later see. The DBLP metamodel is shown in Figure 3. It corresponds to the XML structures used to store the data in the DBLP system.

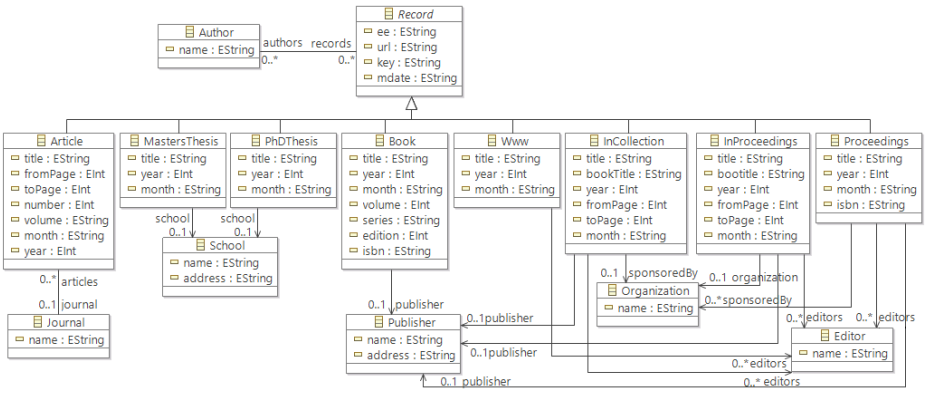


Fig. 3. DBLP metamodel.

Finally, there is the need to incorporate information about the quality of the contributions, in terms of how they rank in commonly used international

reports, such as JCR for journals and CORE for conferences. This information is captured by the Ranking metamodel shown in Figure 4.

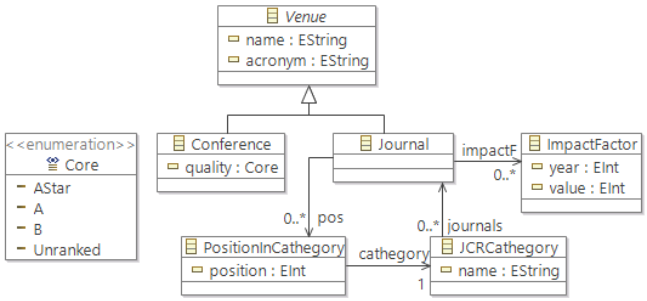


Fig. 4. Ranking metamodel.

2.2 Extracting the data

The second step consisted of populating the models with the data from the three different sources.

Getting the DBLP information was quite straightforward. An XML file with all the DBLP data can be downloaded from the DBLP website (`http://dblp.uni-trier.de/`). Once downloaded, it is a matter of parsing it to generate a model conforming to the DBLP metamodel previously shown in Figure 3. There are two different strategies when parsing an XML file: event-based, where that the parser launches an event each time a starting or an end tag is found during the parsing process; and tree-based, where the parser loads the entire XML document, generates the tree and once it is complete it is returned. Given the size of the XML file, performance and the memory usage was a critical factor therefore, our choice was an event-based parser – in particular we used SAX, which better fits our needs. Another alternative could have been the use of the API that EMF offers to handle XML files. However, the performance of SAX was a key factor for selecting it to parse the large XML files.

The information about the members of the community came from SISTEDES directly. We were provided with an Excel file that contained the information described in the metamodel shown in Figure 2 for 308 members.

Finally, the ranking information about the journals and conferences came from two different sources. From an Excel file with the 2015 edition of the JCR listings (the latest available JCR report at the time of the study) we extracted the ranking information about the journals. Given that our scope was the scientific production in the context of SISTEDES, we only considered three JCR categories: Software Engineering, Information Systems, and Theory and Methods. Although the members of SISTEDES also publish in journals from other

categories (Medical information, Multidisciplinary, etc.), we focused in the journals that fall within the scope of the Society.

Conferences were ranked using the information taken from a SCIE[1] report on Computer Science conferences, which used several international rankings, including CORE, and calculated impact factors for the most relevant conferences. A total of 201 conferences were considered in this study, covering all areas of potential interest to SISTEDES (software engineering, databases, logic, programming, etc.). Conferences were classified in three groups, `A*`, `A` and `B`, depending on their calculated impact factor according to that report. The spreadsheet with the ranking of conferences is available from `https://www.dropbox.com/s/2a85fuki3zdx9dw/RankingCongresos2015.pdf?dl=0`. From that file the model with the conferences ranking was populated.

2.3 Merging all sources of information

The information from the three sources needed to be merged into a single, integrated model. For this we designed a Unified metamodel (shown in Figure 5) that contains all the relevant information for our purposes.

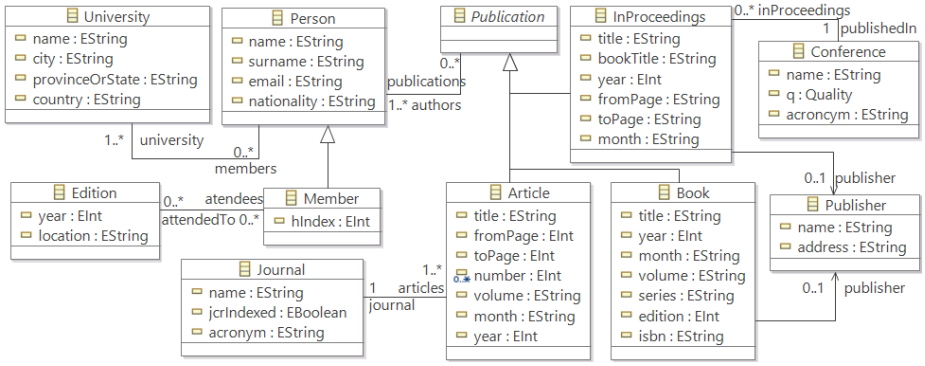


Fig. 5. Unified metamodel.

Then, it is just a matter of developing a model transformation that integrates the models with the three sources of information into one consolidated model conforming to the Unified metamodel. One of the problems found here is that the size of the DBLP model, and hence of the unified model, was too large to be handled by existing model transformation engines, such as QVT or ATL. Then, we decided to use jLinTra [3,4], a parallel model transformation language and engine that is also able to deal with very large models (no matter if they can fit in memory or need to reside in external storage).

[1] Sociedad Científica Informática de España, `http://www.scie.es`

The size of the DBLP model that we used is 5 millions elements, basically the same as the size of the resulting unified model. The transformation that builds the consolidated models took 132.88 seconds in a machine with Ubuntu 12.04 64-bits operating system, 11.7 Gb of RAM and 2 processors with 4 hyperthreaded cores (8 threads) of 2.67GHz each. The transformation contains 732 lines of Java code, and was built in 5 hours by an experienced developer.

Another interesting problem had to do with establishing the *correspondences* between the different models. We decided to use the names of the members to match the names of the authors in DBLP, and the acronyms of the journals and conferences to match the DBLP and ranking models (i.e., name-based correspondences). This implied that we had to used normalized information, making sure that the names of the authors in DBLP and the names of the members referred to the right person. This process took some time, because both the community and the ranking models had to be manually checked for consistency.

3 Queries, Results and Discussion

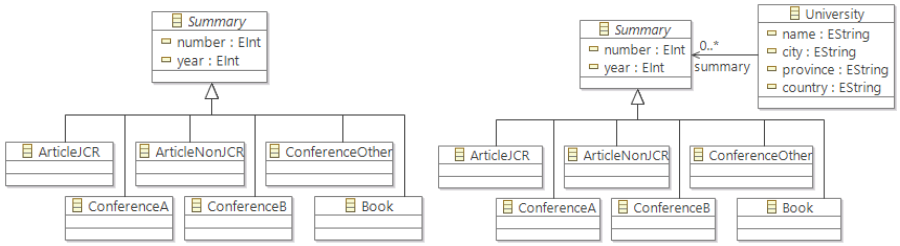
Once we have a model with all the information that we need, it is just a matter of performing the queries to generate the figures for the report. Basically, the kind of queries consisted in extracting information about different aspects of interest to the proponents of the report. For example, the number of contributions per year of the community, classified according to the kind of publication (JCR journal, non-JCR journal, A-conference, B-conference, etc.). We could also ask for that information, but organized according particular subgroups such as individual universities, or considering only quality publications (JCR journals and A*-conferences). Moreover, we could query the model to identify the journals and conferences that are normally used by the members of the community to disseminate their results.

Queries were modeled by developing a metamodel for each one, and defining a transformation from the unified model into them, which produced the required results. Figure 6 shows two of the metamodels used to generate the results for two of the queries.

The transformation was also developed in jLinTra. It contains 323 lines of Java code, and was built in 4 hours by an experienced developer. Starting from every member of the community, it navigates the unified model in order to extract the information about his/her publications and the university where he/she belongs. Then, it synthesizes the results obtained in order to determine the number publications of each type.

The execution of the queries using jLinTra took 3.644 seconds. Given the parallel nature of LinTra, executing in parallel one model transformation for each query, or executing a single model transformation that produced the output models of all queries yielded very similar results.

Finally, the resulting models were parsed and transformed into csv files for the corresponding exploitation to produce the report. Nevertheless, in the future we plan to do this by means of a model-to-text transformation.



(a) Query 1. (b) Query 2.

Fig. 6. Metamodels for two of the queries.

4 Conclusions and Further Work

In this paper we have described the "Making of" of a report about the scientific production of the SISTEDES community, using DBLP as bibliographic source [2] We have shown how the information can be organized, merged and exploited using MDD techniques, and a technical solution is valid and feasible. We have also realized that existing MDD languages and platforms still find problems for coping with large models, but recent proposals (such as LinTra) permit supporting the scalability and performance required to address some of these issues.

Using MDD solutions also brings along interesting advantages, and in particular modularity. One important benefit of the architecture of the solution, depicted in Figure 1, is that it does not necessarily depend on the specific sources. In fact, we have worked in the report for the production of the SISTEDES community using the DBLP information, but reports for other communities, or using different bibliographic databases, are easy to develop. It is just a matter of defining transformations from the specific sources to the metamodels that are used as input in our merging transformation, but whole process does not need to be changed. Similarly, generating new queries becomes a simple matter of developing a model transformation from the unified metamodel to the metamodel of the query of interest. Thus, for example it would be interesting to compare the results obtained using DBLP with the ones that we would obtain if we used the Google Scholar database, or SCOPUS.

Furthermore, this work is a preliminary step into a more detailed study on the exploration of communities using MDD techniques. We plan to extend our study according to different research lines. First, the description of the members of the community is currently achieved in an explicit manner, i.e., giving the list of members. But sometimes we would also like to explore the bibliographic source using wildcards or any implicit description method (e.g., all Spanish authors). This is difficult in general, mainly because the information about the authors in DBLP is quite limited, which means having to provide the required informa-

[2] The report will be available from the SISTEDES website (`http://www.sistedes.es/informes-recursos/`) in September 2015.

tion about the authors using alternative mechanisms. A second improvement of our work would consist of using weaving models between the different sources instead of the very basic name-matching approach used here. Thirdly, we want to conduct a comparison analysis between our proposal and other approaches to bibliometric analysis that use traditional methods (e.g., databases). Similarly, the use of JLintra allowed us to handle very large models in an efficient way, but there are other approaches that use solutions such as EMFStore, CDO, Morsa or Neo4EMF and they provide querying mechanisms. A comparison analysis with these approaches could also help us better understand the advantages and limitations of our proposed solution. Fourth, we would like to automate the report generation and come up with a Domain Specific Language so that users could define the reports they want/need, and the whole process could take care of everything, specially the generation of the queries metamodels. Finally, we would also like to explore our points in common with MetaScience [5], a recent tool to analyze Scientific Conferences. Although the focus of the tool is on conferences and not on communities, many of the ideas of the two proposals could be shared.

Acknowledgments This work is funded by Research Project TIN2014-52034-R and by the Universidad de Málaga (Campus de Excelencia Internacional Andalucía Tech).

References

1. OMG: MOF QVT Final Adopted Specification. Object Management Group. (2005)
2. Jouault, F., Allilaire, F., Bézivin, J., Kurtev, I.: ATL: A model transformation tool. Science of Computer Programming **72**(1-2) (2008) 31–39
3. Burgueño, L., Troya, J., Wimmer, M., Vallecillo, A.: On the concurrent execution of model transformations with linda. In: Proc. of BigMDE'13, Budapest, Hungary, ACM (2013) 3:1–3:10
4. Burgueño, L.: Concurrent Model Transformations based on Linda. In: Doctoral Symposium @ MODELS. (2013)
5. Cánovas, J., Consentino, V., Cabot, J.: MetaScience – a tool to analyze research conferences (2015) `http://atlanmod.github.io/metaScience/`.