Exploring Interface Paradigms for Information Models Mapping

António Painha[1,2], Hugo Manguinhas[1,2], José Borbinha[1,2]

[1]INESC-ID, Rua Alves Redol 9, Apartado 12069, 1000-029 Lisboa, Portugal
[2]IST – Department of Information Science and Engineering, Instituto Superior Técnico, Lisbon Technical University, Portugal
{antonio.h.painha, hugo.manguinhas, jlb}@ist.utl.pt

Abstract. The mapping of information models has become relevant to enable the reuse of structured data (i.e. information) in different contexts. It often resorts to visual mapping tools as the simplest and fastest way to define the needed relationships between two different models and create the desired mapping model. However, the current approaches adopted by visual mapping tools to visualize the mappings between two models have limitations when the models or the mappings become large. This work proposes six new, or improved, visualization techniques (called interface paradigms) to address this problem, aiming to simplify the viewing, making it possible for the user to effectively deal with large models and maps. These proposals were integrated into the XMApper prototype, a visual mapping tool. A user study was conducted to evaluate the proposed interface paradigms, and results led to the conclusion that most of the new paradigms are very useful and effective, when applied to the interface of a visual mapping tool. The primary contribution of this work is a demonstration of new ways to effectively present highly complex mapping information.

Keywords: Visual Mapping Tools, Visualization Techniques, Visualization of Complex Information, Visual Interfaces, XMApper.

1 Introduction

Mapping between complex information models can be a very difficult task [1]. Current well-received solutions to this problem are visual mapping tools [2]. However, a problem with such applications is their inability to cope well with the growth in size and complexity of both the models and the mappings that can be established between them. In the scope of an initiative to develop a web-based Metadata Registry (MDR) [3], a solution to assist and promote a common understanding of the information managed within an organization and in scenarios of interoperability, we developed the XMApper visual mapping tool (Fig. 1).

In XMApper, six new, or improved, interface paradigms were designed and tested in a user study. The study results revealed a significant time advantage for using the new paradigms over the baseline version. In addition, user satisfaction ratings corroborated those performance results.

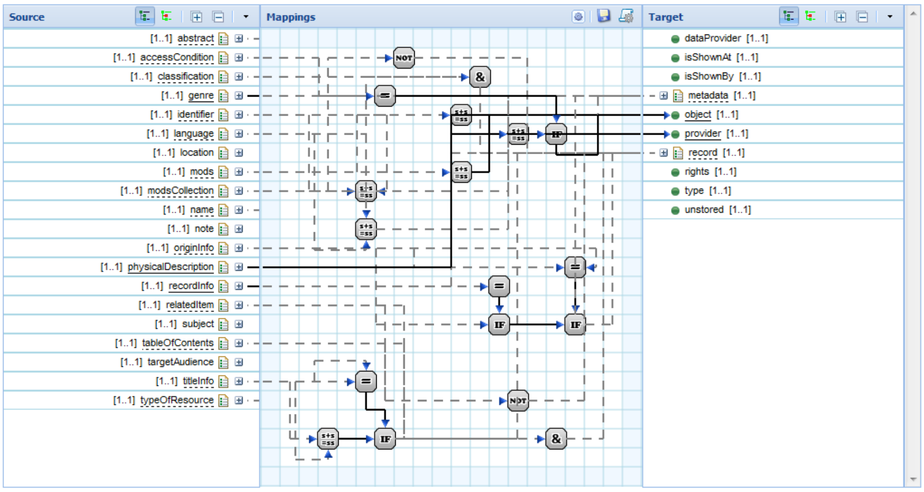

Fig. 1. The XMApper's base interface (the bottom connection that appears to be solid-and-grey is in fact two overlapping dashed-and-grey connections).

2 Proposed Interface Paradigms

After studying the interface paradigms (for large mapping visualization) of modern visual mapping tools (MapForce[1], BizTalk Mapper[2], XML Mapper[3], Clio[4] and COMA++[5]) it was determined that some problems still remain. On the Mapping Board, correct identification of connections, without selecting them, remains the biggest issue, while in the Schema Views the main problems are finding and viewing specific information about the models and relevant information about the mappings. This study and its conclusions led to the idealization of four new and two improved interface paradigms to approach the issues of large mappings visualization, divided between Mapping Board Improvement and Schema View Improvement (paradigms referenced as **[SP1.1]** and **[SP2.2]** are improvements on already existing ones, while the other four are new proposals). Here a small description of each shall be given:

- **Mapping Board Improvement:**
  - **[SolutionParadigm1.1] Connection Visibility –** Provides different visibilities for the mapping connections: when both ends are visible will be drawn solid and black; dashed and gray with only end visible; hidden when no end is visible.
  - **[SP1.2] Connection Render –** Allow the user to change how connections are rendered at any time, providing different views over the same mappings for better element identification and general reading.

---

1 http://www.altova.com/mapforce.html
2 http://msdn.microsoft.com/en-us/library/ee253382(v=bts.10).aspx
3 http://www.stylusstudio.com/xml_mapper.html
4 http://www.almaden.ibm.com/cs/projects/criollo/
5 http://dbs.uni-leipzig.de/Research/coma.html/

- **[SP1.3] Connection Argument Visibility –** Show the inputs and outputs names of the mapping functions as floating text above the end of the connection linking to them. Allow the user to configure when these connection arguments appear.

- **Schema View Improvement**
  - **[SP2.1] Quick Filters –** Quick access, by toggles, filters that allow the user to isolate important elements or show a view of interest in the Schema View. For example, one of these filters hides all unconnected elements.
  - **[SP2.2] Tree Element Connection Status –** Use a connection line and a text underscore in the Schema View elements, to give information of the connection status of the element to the user.
  - **[SP2.3] Sorting –** Permits the sorting of the Schema View elements, to help localization in heavy populated schema views. Two relevant types of sorting: alphabetical and by number of connections.

3 User Study

A user study was conducted to evaluate the usability and usefulness of the developed solution. Ten computer users, with various levels of experience regarding visual mapping tools, were invited to test three different versions of the prototype, incrementally implementing more paradigms (namely, Version A – baseline version; Version B – version A plus the Mapping Board Improvement paradigms; and Version C - version B plus the Schema View Improvement paradigms. For all versions, even the baseline one, the **[SP2.2]** Tree Element Connection Status was active due to its fundamental value for identifying connected schema elements). The tests consisted in all participants completing appointed task sets with all three versions of the interface. To control for order effects, the order in which participants experienced each version of the mapper was counterbalanced using a Latin Square[6] design. After all the prototype tests were done, each participant completed a background and satisfaction survey.

Four task sets with four tasks each were devised, involving finding elements and information in the source and target schema, their related function boxes and connections. An effort was made to keep the task sets isomorphic so that the participants experienced similar tasks as they viewed each version of the interface. To ensure that no one task set was accidentally more difficult that the rest, however, they were rotated through visualizations. The map used in the experiment is shown in Figure 1.

Figure 2 shows the results of the post-hoc analysis of all obtained task times and user satisfaction ratings and their relative standard deviations (σ).

The average task time for the prototype Version A (base version plus **[SP2.2]**) was significantly higher than that of Version C (all paradigms), being 28.4 (σ=3.2) and 17.9 (σ=4.0) seconds respectively. Version B (baseline plus **[SP1.1]**, **[SP1.2]** and **[SP1.3]**) with an average task time of 22.1 seconds (σ=3.2), stood in the middle of the other two. Interaction between prototype version and tasks revealed that task 1 was the hardest, and task 3 and 4 were especially difficult when using Version A.

[6] http://mathworld.wolfram.com/LatinSquare.html

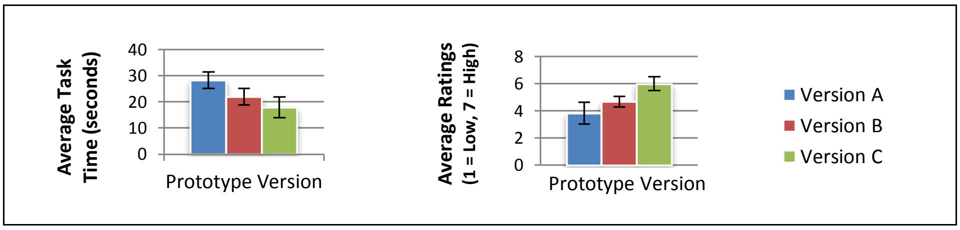

Fig. 2. Task times and user satisfaction ratings for each of the three prototype versions.

A user satisfaction questionnaire was completed by the participants at the end of the session. To improve the methodological rigor, some statements were asked in a favorable way toward the prototypes tested and some were phrased in a negative manner. Responses were collected using a 7-point Likert scale with 1 = Very Low and 7 = Very High. In order to improve readability, questions which required a lower response to reflect a positive satisfaction were flipped prior to analysis (e.g. if the user rated a question with 1, meaning the highest possible value, it was flipped to 7). Version C was rated significantly higher than the others, and Version B was slightly higher rated than Version A. The standard deviations were low for all versions (A: σ=0.8; B: σ=0.4; C: σ=0.5), representing a good accuracy for the results.

4 Conclusion and Future Work

Addressing the visualization issues created by large mappings was the goal of this work. Four new and two improved interface paradigms were designed and tested in a user study. Overall results demonstrate that, although being clearly a work in progress, the new paradigms are valid in helping the visualization of large mappings, even if some better than others.

There is still much to do to improve and expand the new interface paradigms. Some examples: more connection renders; more quick filters can be included in tree drop down menu, and choosing those that can be accessed by toggling should be possible. Test participants also suggested several interface changes for the paradigms, especially in their accessibility.

References

1. Robertson, G., Czerwinski, M., Churchill, J.: Visualization of Mappings Between Schemas. In: CHI'05: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 432-439. ACM, New York (2005).
2. Alexe, B., Tan, W.C., Velegrakis, Y.: Comparing and Evaluating Mapping Systems with Stbenchmark. Proc. VLDB Endow. 1468-1471 (2008)
3. Manguinhas, H., Borbinha, J.: Integrating Knowledge Organization Systems Registries with Metadata Registries. In: The 9th European NKOS Workshop at the 14th ECDL Conference, Glasgow (2010)