Poster: A Bidirectional Model-driven Spreadsheet Environment

Jácome Cunha*, João Paulo Fernandes*†, Jorge Mendes*, João Saraiva*
*HASLab, INESC TEC & Universidade do Minho, Portugal
†Departamento de Engenharia Informática, Universidade do Porto, Portugal
{jacome,jpaulo,jorgemendes,jas}@di.uminho.pt

***Abstract*—In this extended abstract we present a bidirectional model-driven framework to develop spreadsheets. By being model driven, our approach allows to evolve a spreadsheet model and automatically have the data co-evolved. The bidirectional component achieves precisely the inverse, that is, to evolve the data and automatically obtain a new model to which the data conforms.**

***Keywords*-Model-driven Engineering, Bidirectional Transformations, Software Evolution, Model Evolution, Spreadsheets**

I. Introduction

Spreadsheets are widely used in the development of business applications. Spreadsheet systems offer end users a high level of flexibility but this freedom comes at a price: spreadsheets are notoriously error prone as shown by numerous studies reporting that up to 90% of real-world spreadsheets contain errors [1].

In recent years, the spreadsheet research community has recognized the need to support *model-driven spreadsheet development* (MDSD), and to provide spreadsheet developers and end users with methodologies, techniques and tool support to improve their productivity. In fact, several techniques have been proposed, namely the use of templates [2] and *ClassSheet* models [3].

These proposals allow users to define a spreadsheet business model from which a customized spreadsheet application holding the actual data is generated, while the consistency of the spreadsheet data with the overlying model is guaranteed.

A significant drawback of such approaches lies in the fact that the evolution of both spreadsheet models and the instances generated from them is considered in isolation. Thus, a simple evolution step on the model side could break the conformity with its instance, and vice versa.

In an attempt to overcome this drawback, we have proposed in the past a single and coherent environment for MDSD. For this, we have started by embedding *ClassSheets* in a traditional spreadsheet system [4]. Later, we have proposed to automatically co-evolve spreadsheet instances upon a model transformation [5], a feature that has also been integrated in a tool [6]. Finally, we have defined the theoretical setting to reason about spreadsheet models and instances in the reverse direction: we allow for data instance evolutions while models are automatically co-evolved [7].

The purpose of this extended abstract, together with the poster that accompanies it, is to introduce the global bidirectional model-driven spreadsheet environment that we finally achieve.

II. The MDSheet Evolution Environment

In [6] we have presented a spreadsheet system that provides a MDSD environment. Figure 1 shows such a spreadsheet environment for managing a yearly travel budget.

This environment has the following features:

- *ClassSheet* models are embedded in spreadsheet systems [4], providing a coherent environment for MDSD. The environment consists of two worksheets: one contains the *ClassSheet* model (model worksheet), and the other contains the spreadsheet instance (data worksheet), as shown in Figure 1.
- Initial and customized spreadsheet instances are generated from *ClassSheet* models. Such instances include tailor-made versions of update operations.
- The customized spreadsheet instance guides end users in introducing data that conforms to the embedded model: for example, rows with type integer only accept integer values. Another key feature of this approach is that blocks of cells are automatically produced, for example, to add a new year budget, that consists of three columns/attributes, the user must press on the button in column H. Formulas are also automatically updated, in our example the summation formulas in column I.
- Spreadsheet users can evolve the *ClassSheet* model, by using predefined evolution step operations. The MDSheet environment provides buttons (c+, c-, Col+, etc.) that perform typical model evolution steps, like adding a column/row in a *ClassSheet* class (for example, adding a column/attribute taxes to the class year). After each evolution step, the environment automatically co-evolves the spreadsheet data.

This spreadsheet environment[1], however, provides only model evolution and spreadsheet data co-evolution. In the next section we describe the extension of the core evaluation engine of the MDSheet framework to provide a bidirectional transformation mechanism so that spreadsheet users can evolve both the *ClassSheet* model and the spreadsheet data.

[1]A demonstration video of this environment is available at http://ssaapp.di.uminho.pt.

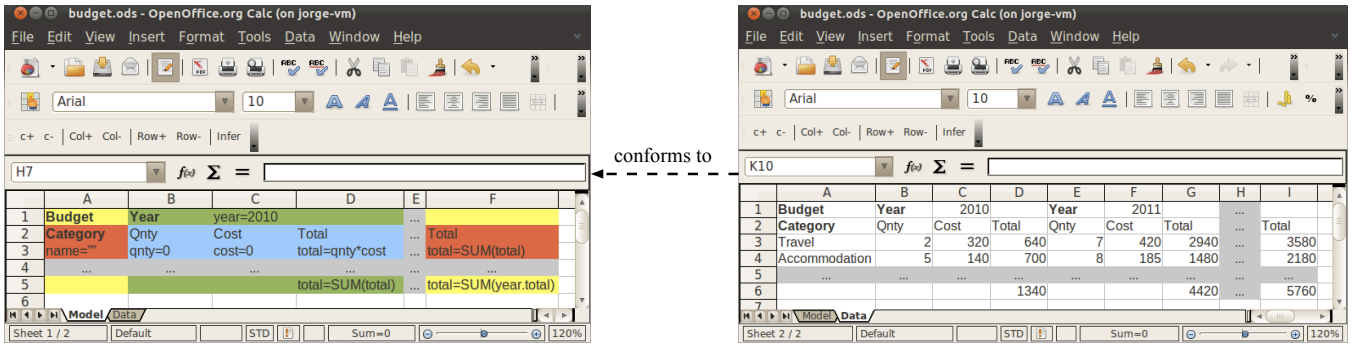

Figure 1: Model-driven spreadsheet environment for a yearly travel budget.

III. A Bidirectional Extension for MDSheet

A bidirectional MDSheet environment extends the features described in the previous section with the possibility of end users evolving both the *ClassSheet* model and the spreadsheet data. The *ClassSheet* or spreadsheet data evolution steps are modeled as model or data transformations.

The *ClassSheet* model transformations are related to a sequence of transformations on the data. This relation ensures that a valid model transformation is equivalent to a valid data transformation and that the evolved data conforms to the evolved model. A similar approach has been done for data transformations by relating them to model transformations as depicted in Figure 2. These relations guarantee safety by construction.

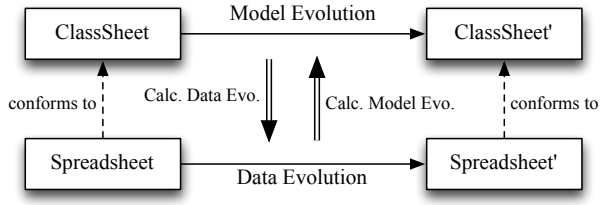

Figure 2: Bidirectional transformational system diagram.

The theoretical foundations of our bidirectional evolution techniques are described in [7], where we also present the API defining all evolution steps available for the *ClassSheet* model and the spreadsheet data.

We have extended our MDSheet addon in order to provide bidirectional evolution mechanisms to the OpenOffice spreadsheet system. This addon consists of a set of OpenOffice Basic macros and visual buttons that provide common evolution step operations on spreadsheet data (and models). In fact, the environment shown in Figure 1 supports bidirectionality: evolution step buttons are available in both the data and model worksheets. For example, the Col+ button in the data worksheet evolves the data by adding a new column in all instances of the selected class. A more detailed description of the architecture of the bidirectional MDSheet environment is presented in the accompanying poster.

References

[1] R. Panko, "Spreadsheet errors: What we know. what we think we can do." *Proceedings of the Spreadsheet Risk Symposium, European Spreadsheet Risks Interest Group (EuSpRIG)*, 2000.

[2] R. Abraham, M. Erwig, S. Kollmansberger, and E. Seifert, "Visual specifications of correct spreadsheets," in *IEEE Symposium on Visual Languages and Human-Centric Computing*. IEEE Computer Society, 2005, pp. 189–196.

[3] G. Engels and M. Erwig, "ClassSheets: automatic generation of spreadsheet applications from object-oriented specifications," in *20th IEEE/ACM Int. Conf. on Automated Sof. Eng., Long Beach, USA*. ACM, 2005, pp. 124–133.

[4] J. Cunha, J. Mendes, J. P. Fernandes, and J. Saraiva, "Embedding and evolution of spreadsheet models in spreadsheet systems," in *IEEE Symp. on Visual Languages and Human-Centric Computing*. IEEE CS, 2011, pp. 179–186.

[5] J. Cunha, J. Visser, T. Alves, and J. Saraiva, "Type-safe evolution of spreadsheets," in *Int. Conf. on Fundamental Approaches to Software Engineering*, ser. FASE'11/ETAPS'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 186–201.

[6] J. Cunha, J. P. Fernandes, J. Mendes, and J. Saraiva, "MDSheet: A framework for model-driven spreadsheet engineering," in *ICSE'12: Proceedings of the 34rd International Conference on Software Engineering*. ACM, 2012, to appear.

[7] J. Cunha, J. P. Fernandes, J. Mendes, H. Pacheco, and J. Saraiva, "Bidirectional transformation of model-driven spreadsheets," 2012, submitted.