

---

# **MDSheet for Google Sheets Grant Report**

**(Ref.: BI1-2013\_PTDC/EIA-CCO/120838/2010\_UMINHO)**

**Jorge Mendes**

**jorgemendes@di.uminho.pt**

---

**Techn. Report TR-HASLab:04:2014**

**December 2014**

---

**FATBIT – Foundations, Applications and Tools for Bidirectional  
Transformation  
(Project FCOMP-01-0124-FEDER-020532)**

**HASLab - High-Assurance Software Laboratory  
Universidade do Minho  
Campus de Gualtar – Braga – Portugal  
<http://haslab.di.uminho.pt>**

---

**TR-HASLab:04:2014**

*MDSheet for Google Sheets*

*Grant Report*

*(Ref.: BI1-2013\_PTDC/EIA-CCO/120838/2010\_UMINHO)*

by Jorge Mendes

**Abstract**

This technical report describes the research goal and the result obtain for the research grant with ref. BI1-2013\_PTDC/EIA-CCO/120838/2010\_UMINHO and supervised by Prof. João Saraiva, within the FATBIT research project.

The work herein described aims to take MDSheet's features to an online spreadsheet development environment. For that, Google Sheets was extended with features to design and work with MDSheet models.

In the end, the undertaken approach is not ideal for end users' production systems, due to latency effects introduced by the used architecture. Nevertheless, this still is a valid approach, but that would require support from Google to improve the architecture and thus reduce the flaws in the current one.

---

# 1 Introduction

In the context of this research grant, it was defined to develop techniques to evolve model-driven spreadsheets [2] in an online collaborative environment. Previous work in the area includes the embedding of spreadsheet models, namely *ClassSheets* [4], into spreadsheets themselves [3]; the specification of transformations on both models and respective instances [1]; and the development of an OpenOffice/LibreOffice extension implementing these techniques [2].

For this work, another possible approach was proposed using a different spreadsheet host system. This system should be able to provide a highly collaborative environment, thus an online web-based environment was preferred. From the available systems meeting this criterion, *Google Sheets* was chosen, since it provides online collaborative spreadsheets with real-time collaboration with the ability to be accessed in distinct ways, including via its web interface and a public API.

Google Sheets is a simple spreadsheet system, with a reduced set of features when comparing with more complex systems like Microsoft Excel and OpenOffice Calc, but it provides the features required for basic spreadsheet usage. These characteristics are enough to develop a prototype with some of MDSheet's features.

The description of the prototype's architecture follows.

# 2 Architecture

The architecture of the MDSheet-Google system can be decomposed in two sub-systems: the user interface, and the MDSheet server. The former is responsible of the interaction with the user and makes the necessary calls to the MDSheet server, whilst the latter receives these calls and operates on the received models and instances. Moreover, the MDSheet server is only a wrapper around the core of MDSheet, providing web services for accessing its functionalities. The communication between the user interface and the server is based on REST requests from the user interface to the server, where the messages are represented in JSON (see figure 1).

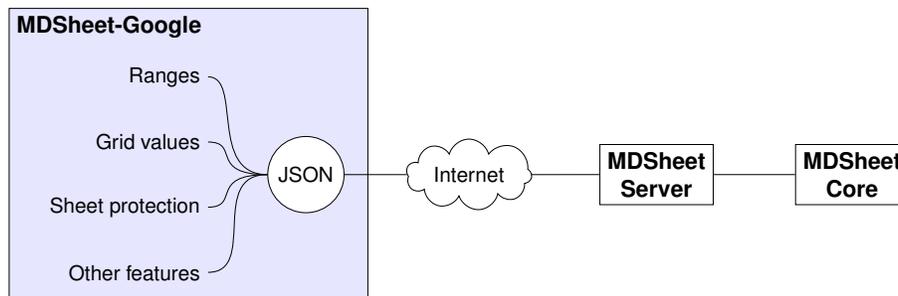


Figure 1: Architecture of the MDSheet-Google system.

The user interface (UI) for MDSheet is implemented as a Google Sheets add-on. It implements an embedding of MDSheet's ClassSheet-based models within Google Sheets and provides the tools for users to make use of MDSheet's features. When a user performs an action that makes use of any MDSheet feature, e.g. setting the value

of a cell or adding a new column, the UI sends a request to the MDSheet server, serializing the needed data to JSON. The MDSheet server, when receiving a request, parses the content performing some basic validations and then calls the respective function in the MDSheet API.

The communication between the user interface and the MDSheet server is performed using common HTTP POST requests. The requested path indicates which action is to be performed, and the Uniform Resource Identifier (URI) query part includes the arguments for the operations. The values on which the actions are to be performed are sent in the content of the POST request, represented in JSON.

### 3 Conclusion

Although possible, the use of model-driven spreadsheet development in an online web-based collaborative environment is not yet completely feasible for end users' production environments. Some of the shortcomings of this system include (a) the lack of more advanced features available on common and more powerful spreadsheet systems, e.g., Microsoft Excel and OpenOffice Calc; (b) the requirements for end users' systems to access the spreadsheet web interface and run the JavaScript software, which can be slow on common machines; and, (c) the network characteristics between Google Sheets servers and the MDSheet web service server.

For (a), new features can be added to Google Sheets using extensions, as done with MDSheet, but users need to rely on third parties. For (b) and (c), the system could be improved with enhancements on Google's side, with the integration of the UI for MDSheet into the Google Sheets interface, and the use of Google Sheets servers to run MDSheet in order to reduce network traffic and control latency and response times when performing operations.

### References

- [1] Jácome Cunha, João P. Fernandes, Jorge Mendes, Hugo Pacheco, and João Saraiva. Bidirectional transformation of model-driven spreadsheets. In *ICMT '12*, volume 7307 of *LNCS*, pages 105–120. Springer, 2012.
- [2] Jácome Cunha, João Paulo Fernandes, Jorge Mendes, and João Saraiva. MDSheet: A framework for model-driven spreadsheet engineering. In *Proc. of the 34rd Int. Conf. on Software Engineering (ICSE)*, pages 1412–1415. ACM, 2012.
- [3] Jácome Cunha, Jorge Mendes, João Paulo Fernandes, and João Saraiva. Embedding and evolution of spreadsheet models in spreadsheet systems. In *IEEE Symp. on Visual Lang. and Human-Centric Comp.*, pages 186–201. IEEE, 2011.
- [4] Gregor Engels and Martin Erwig. ClassSheets: automatic generation of spreadsheet applications from object-oriented specifications. In *Proc. of the 20th IEEE/ACM Int. Conf. on Aut. Sof. Eng.*, pages 124–133. ACM, 2005.