

The Simplified Platform, Status 2023

Mark A. T. Mulder¹, Rick Mulder

¹TEEC2, Hoevelaken, the Netherlands

Abstract

Simplified is the web based approach to modelling and meta-modelling. This platform started from the experience with a previous research tool for modelling Design and Engineering Methodology for Organisations (DEMO) and has increased the available notations to OntoUML and ArchiMate. The extension ability of the cloud based platform makes it suitable for research and business applications. The configurable notations, flexible user interface, and real-time transformation, verification and visualisations makes the platform adaptable and understandable for every stakeholder. This update paper will list the current state of the simplified platform.

Keywords

Modelling, Meta-modelling, Collaboration, Enterprise Engineering

1. Introduction

The history of the Simplified modelling platform started with the research project towards the PhD ‘Enabling the automatic verification and exchange DEMO models’ [1]. The DEMO [2, 3] method is a core method (based on a theoretically founded methodology) within the discipline of Enterprise Engineering (EE) [4]. We have described the history in detail in our previous paper [5].

The lack of good tooling for demo modelling prompted us to start the development of a new tool. Via the plugin on Sparx Enterprise Architect we started development on a cloud based modelling platform which supports collaborative design, multiple notations, API and white label UI integration that would allow customers to apply their own corporate design language to the UI, and multiple languages all while applying state of the art development methods.

This paper describes the used architecture, the implemented interfaces and the current status of the platform. After that a series of features are introduced and respective benefits for research, business users of organisations, and modellers in general are introduced. We also provide some evaluation procedures used to get feedback on the platform usage. Finally, we conclude with a short summary and the current state of affairs of this platform.

Companion Proceedings of the 16th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling and the 13th Enterprise Design and Engineering Working Conference, November 28 – December 1, 2023, Vienna, Austria

✉ markmulder@teec2.nl (M. A. T. Mulder)

🆔 0000-0002-1846-0238 (M. A. T. Mulder); 0000-0002-9500-0702 (R. Mulder)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

2. Architecture

The platform consists of total of six layers, divided in two servers: application server (involving the layers interface, message, process, cache, and persistence), and database server (database layer). The server architecture is visualised in fig. 1. The layers are visualised in fig. 2

The interface layer consists of two interfaces for accessing the platform. The REST/JSON API is the simplest interface, allowing the retrieval of the generic public information that requires no authentication. No modelling information can be exchanged using this API. This information includes the installed public notations (e.g. ArchiMate, DEMO, BPMN) and the creation of free accounts. The other interface, that facilitates the access to the platform is an authenticated asynchronous web socket messaging interface that can not only receive and handle messages, but also can broadcast to all relevant connected users. This interface can be used by authorised developers to build their own user interactions. These interactions can involve generation, layouts or manipulation of models and model elements.

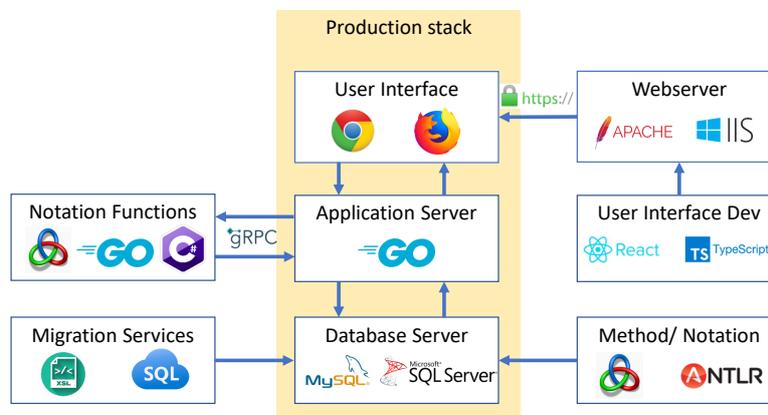


Figure 1: Architecture of Simplified

Messaging to the asynchronous interface is done in JSON format with a dynamic payload structure, allowing for per message configuration, and developing structures during the lifetime of the platform. Other clients can use this structure to interface with the back-end engine for notation specific operations. Every client, that is also a server, can request the platform to forward special operations upon user request. This information is then processed and returned to the client UI for the visualisation.

The modelling part of the back-end is designed to store the model and the metamodel of a notation or methodology. The architecture uses dynamic metamodels that restrict the models on run-time. The notation architecture structure is visualised in fig. 3.

The platform is now available on <https://simplified.engineering/> and the beta-test of new features is running on <https://acc.simplified.engineering>.

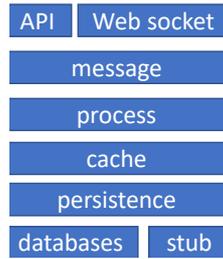


Figure 2: Architecture Layers

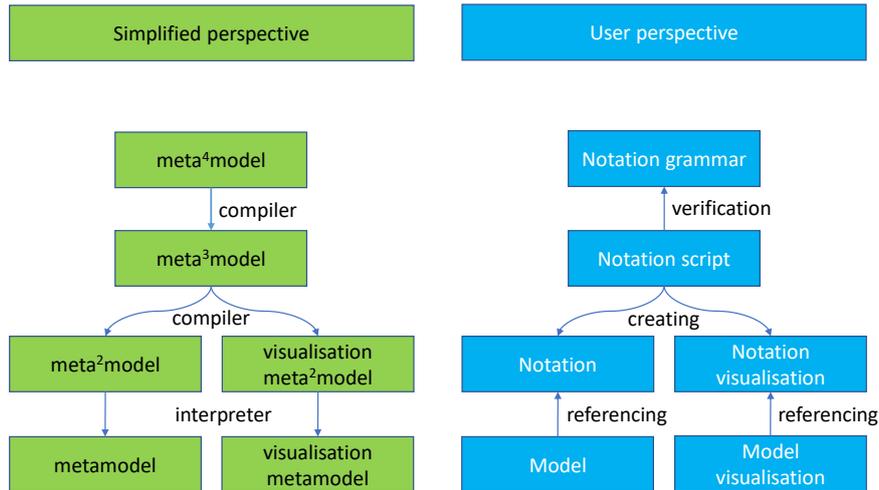


Figure 3: Notations from different perspectives

3. Features and benefits

We observe that some of the features mentioned below should not be regarded anymore as distinctive tool features in the current state-of-the-art because many tools on the web have this same set of features [6, 7, 8, 9, 10, 11, 12].

Next to the baseline features mentioned in the previous paper [5], we have added the following features:

- Collaboration between users of different companies
- Connecting Connections to Connections
- Connections have anchor points to shape the route they take
- Connection quick menu
- Hiding connections
- Improved the ease of creating notation scripts
- Ability to model Elements onto other Elements (e.g. Attribute on an Entity) with its own unique visual
- Adding custom properties to Model Elements and Connections

- Switch notation visuals based on properties
- Align and Rotate functions
- Verification of model based on the notation script
- Server Extensions that can hook into the UI.
- Anchors inside visuals to control certain parts of a visual
- Settings to customise the look of your modeller, such as toolbox icon sizes, grids, snapping and more

We will go over some of these new features in more depth in the next paragraphs

All users working on the same models, and, therefore, automatically collaborate. Next to the elements there is a quick menu which shows connections defined in the notation that are available from this element. With Simplified one can support one's own modelling methodology and associated notation. Elements can be added in the script with the phrase "element < name > (< attributename >< type >,...)" and connections like "connection < name > (< attributename >< type >,...) from < sourceName > (< attributename >< type >,...) to < targetName > (< attributename >< type >,...)"[13]. Furthermore, predefined notation scripts for DEMO, OntoUML, BPMN, ArchiMate, and VISI are uploaded on the platform. Other notations such as PetriNet, ER, UML, etc. can be added on request.

Notation visuals can have defined anchors to move around shapes inside the visual. Quick settings located in the area that concerns them for quickly toggling or changing specific things. This interface allows for cross-programming-language development and we have used it to let remote servers subscribe to our server to link to actions in the user interface. For example, model verification logic can be imported and run against the model to check for its compliance with the underlying notation.

We have four standard and one custom licence types to be available to the users of Simplified. Standard licence offering will include the Basic, Standard, Pro, Enterprise, and Research&Education versions ¹. It is worth noting that Basic licence type will provide a free, thus low barrier entry point to the modelling environment with basic functionality to the user. We categorise Basic and Standard for personal, and the others for professional use.

4. User interface

Simplified is a server based solution which supports models and visualisations of those models [1]. Therefore, new technologies that want to connect to those visualisations can do so by using the messaging system. In the end, the visualisation can be summarised by a view with visual elements having a x,y,z coordinate and a size. Those properties are supported and can be extended within the platform.

Simplified Web UI is built of (web) pages in a modular fashion. Every set of features has its own page. Each page is built with modular function blocks which share a state with the other modular function blocks. Each modular block has its functions and a visual representation. Next, the modeller page consists of the visualisations of a modular function block that have a docking possibility within the page. In addition to sharing a 'session' state between some pages,

¹<https://teec2.nl/products/modelling-platform/>

pages have their own states as well. As explained in the next sections, examples of modular blocks are Toolbox, Browser and Properties. The page layout is saved as a session variable, which allows each user to have its own unique layout. Due to the modular nature, it is also possible to have stakeholder layout presets. Other pages that do not have the docking layout, still use the modular setup, which makes it possible to customise each page to specific needs of the stakeholder.

The browsers will all have the up-to-date data displayed in their own way as shown in fig. 4.

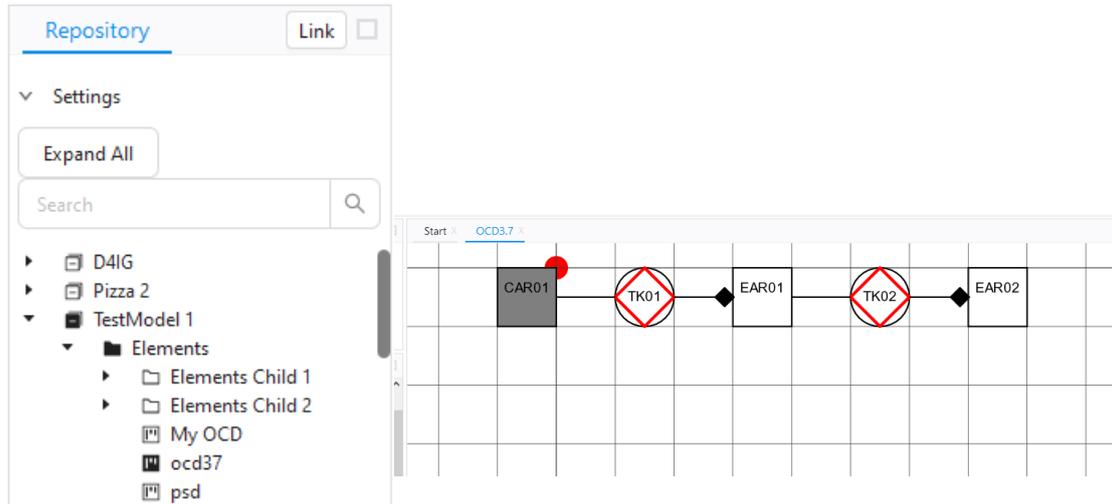


Figure 5: The diagram view

Figure 4: The repository browser

The diagram is arguably the most important part of the modeller. A diagram can be opened from the browser and represents a graphical view or visualisation of a part of the model following the defined viewpoint of the used notation as presented in fig. 5. This allows for smooth collaboration and modelling together.

The toolbox contains the notations that are available to the user as visualised in fig. 6. These notations have categories, of which two are standard, namely connections and elements. The toolbox can be refreshed, which comes in handy in case of an updated notation or a new notation that is not yet shown in the toolbox.

The property module displays the properties of the selected element as visualised in fig. 7.

5. Evaluation

Currently we are developing and beta-testing the DEMO and OntoUML notation and adjusting the user experience of these notations in the Simplified environment. The evaluation is done with students from different universities in their bachelor or master studies. Feedback from users is processed through a ticket system. All reports will be collected and categorised to bugs, improvements and feature requests. The bugs and improvements will be taken care of right away, and will be pushed onto an internal test server. Once these are tested and found to solve

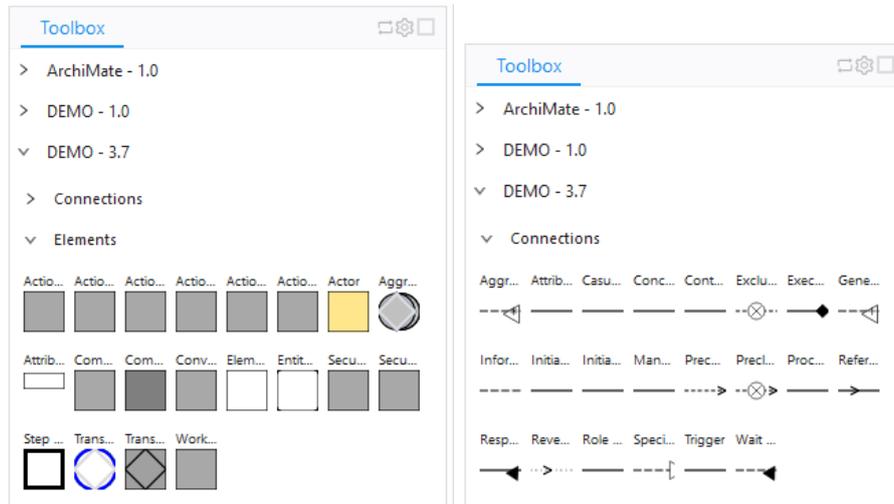


Figure 6: Toolbox with several notations and their categories

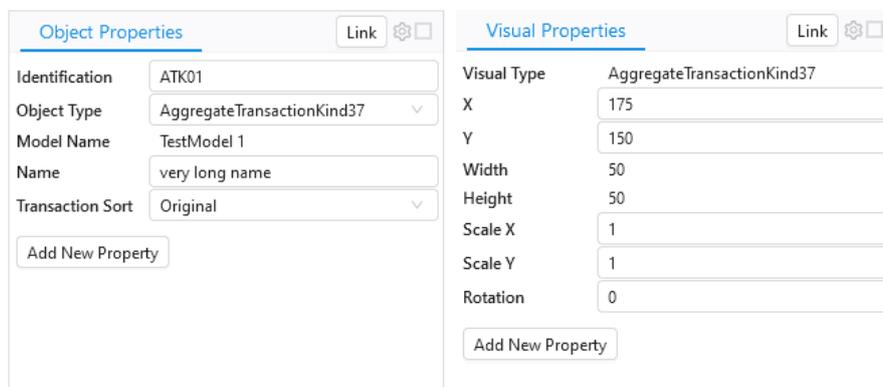


Figure 7: The Element and Visual Element properties

the issue they are meant to fix, they are pushed to the beta-test server and users can work with it. The feature requests will be added to the next iteration of the platform.

6. Conclusion

The platform, Simplified, takes away the limitations that we experienced during the development of Plena. It supports the collaborative design and multiple notations expressed in multiple languages.

Besides, up till now no restrictions have been seen that would impede the addition of more notations to the platform. The extension of features can be done in a modular fashion, allowing for future feature development without disruption.

Additional research is needed to extend the features of modelling and a broader investigation

of the limitations and gaps of other modelling tools is yet to be conducted. We have a road map of features that include, but are not limited to the following features:

- The referencing of models from a template, inheritance (advising or forcing) references.
- The support of visualisation concepts like swim lanes on diagrams, matrices, cubes, and automated layout features.
- User support on choosing the next step of the methodology.
- Advanced ruling that can efficiently verify models.
- Automated transformation between notations, and generation within notations and to other systems.

Finally, a lot of features that come with specific modelling methodologies will be added.

References

- [1] M. A. T. Mulder, Enabling the automatic verification and exchange demo models, 2022.
- [2] J. L. G. Dietz, Enterprise Ontology – Theory and Methodology, 2006.
- [3] J. Dietz, H. Mulder, Enterprise ontology: A human-centric approach to understanding the essence of organisation, 2020.
- [4] J. Dietz, J. Hoogervorst, A. Albani, D. Aveiro, E. Babkin, J. Barjis, A. Caetano, P. Huysmans, J. Iijima, S. J. V. Kervel, The discipline of enterprise engineering, 2013.
- [5] M. A. T. Mulder, R. Mulder, F. Bodnar, M. van Kessel, J. Gomez Vicente, et al., The simplified platform, an overview, 2022.
- [6] Lucid, Where seeing becomes doing, <https://www.lucidchart.com/pages/>, 2022. URL: <https://www.lucidchart.com/pages/>.
- [7] Camunda, Web-based tooling for bpmn, dmn and forms, <https://bpmn.io/>, 2022. URL: <https://bpmn.io/>.
- [8] JGraph, Security-first diagramming for teams, <https://app.diagrams.net/>, 2022. URL: <https://app.diagrams.net/>.
- [9] Edrawsoft, Unlock diagram possibilities, <https://www.edrawsoft.com/>, 2022. URL: <https://www.edrawsoft.com/>.
- [10] Google, Webea - new tabbed interface, <https://sparxsystems.com/products/procloudserver/5.0/index.html#9AEC9F32-C964-4795-844E-FB9145DCDE43>, 2022. URL: <https://sparxsystems.com/products/procloudserver/5.0/index.html#9AEC9F32-C964-4795-844E-FB9145DCDE43>.
- [11] Nulab, Teams who diagram together, thrive together, <https://cacoo.com/>, 2022. URL: <https://cacoo.com/>.
- [12] Creately, Business process mapping tool, <https://creately.com/lp/bpm-software-online/>, 2022. URL: <https://creately.com/lp/bpm-software-online/>.
- [13] M. A. T. Mulder, R. Mulder, F. Bodnar, Towards a demo description in simplified notation script, 2023.