

DEMO-4 extension for Sparx EA

Peter Kuipers^{1,†}, Marien R. Krouwel^{2,3,*,†}

¹*EAXpertise, Eindhoven, The Netherlands*

²*Make IT Right, Utrecht, The Netherlands*

³*Trives Solutions, Utrecht, The Netherlands*

Abstract

Sparx Systems Enterprise Architect is a visual modeling and design tool based on the OMG UML standard. The platform supports many industry standards for modeling business processes and software, including BPMN, DMN, and ArchiMate, and is widely used for modeling enterprise architectures. In 2020, DEMO-4 was launched, to model the ontology of an enterprise as a coherent set of models that describe its products and services, processes, information and business rules. As there is little tool support for the creation of DEMO models that can be integrated with other enterprise models, this tool paper reports on the development of an extension (MDG) to Sparx EA to facilitate the modeling of DEMO in Sparx EA. In order to achieve this result, several decisions were made regarding the DEMO meta model and 14 templates were defined to ease the modeling DEMO action rules. The extension is evaluated with several academic and real-world cases and is currently being used by multiple organizations. This extension to Sparx EA has the potential to support further adoption of DEMO within Enterprise Modeling communities.

Keywords

DEMO, Enterprise Modeling, Sparx EA, Enterprise Architecture,

1. Introduction

Sparx Systems Enterprise Architect¹ (Sparx EA) is a visual modeling and design tool based on the OMG UML² standard. The platform supports many industry standards for modeling business processes and software, including BPMN, DMN, and ArchiMate. Sparx EA has a large user community and is being used by many organizations for modeling enterprise architectures.

DEMO-4, launched in 2020, provides a modeling language to model the ontology or essence of an enterprise, as defined in the EE-theories [1]. A DEMO (Design and Engineering Methodology for Organizations) models consists of a coherent set of (aspect) models (see Figure 1) that describe its products and services (transaction kinds), processes, information (fact types), business rules (action rules) and involved actors or actor roles (responsibility and accountability).

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

*Corresponding author.

†These authors contributed equally.

✉ Peter@EAXpertise.nl (P. Kuipers); Marien.Krouwel@Make-IT-Right.nl (M. R. Krouwel)

🌐 <https://eaxpertise.nl/> (P. Kuipers); <https://make-it-right.nl/> (M. R. Krouwel)

🆔 0000-0003-4115-3858 (M. R. Krouwel)

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

¹<https://sparxsystems.com/products/ea/index.html>

²<https://www.uml.org/>

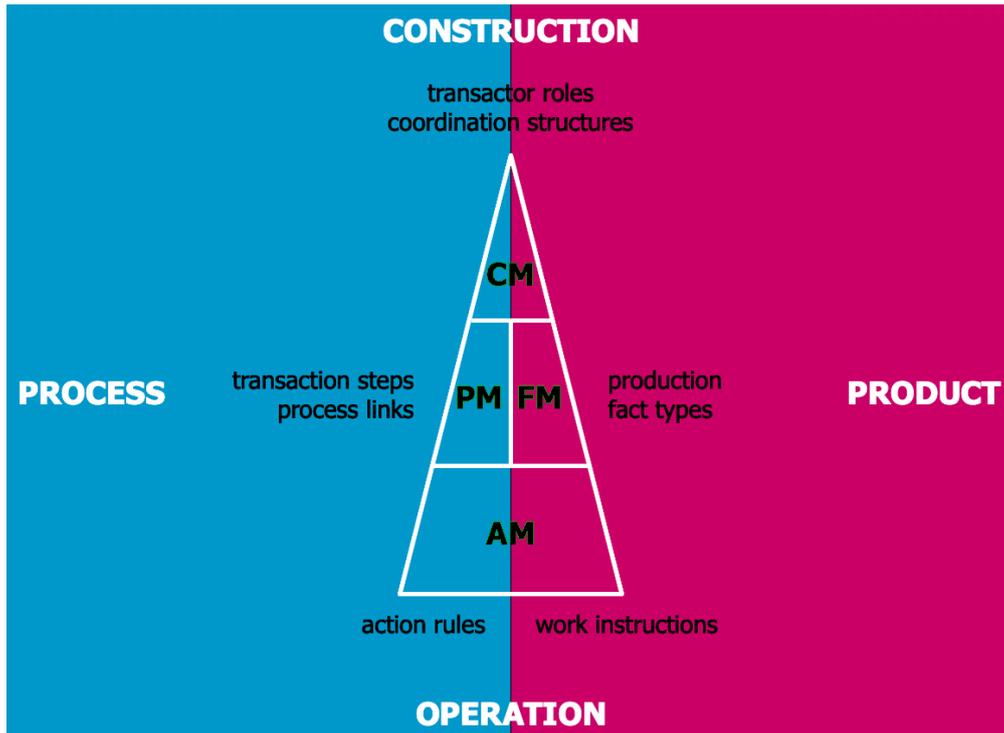


Figure 1: DEMO aspect models [1]

Enterprise models are essential for the understanding and (re)design of an enterprise and its supporting software [2, 3] and can comprise many aspects [4, 5]. Thus, there is a need in the growing DEMO and Enterprise Modeling communities to have a tool that supports the creation of DEMO models that can be integrated with other enterprise models. This tool paper reports on the creation of an extension (MDG) to Sparx EA, a widely used tool for the creation of enterprise modeling, to support the modeling of DEMO in Sparx EA. In the creation of this extension, it was found that the DEMO metamodel was incomplete and inconsistent at some points. Several design decisions have been taken regarding the DEMO metamodel. Additionally, it was found that the modeling of DEMO action rules could be eased by introducing 14 templates. This extension has the potential to support further adoption of DEMO.

This paper is structured as follows: in section 2 the relevance of this tool and the choice for Sparx EA as foundation is outlined, in section 3 a (short) introduction to DEMO (modeling) is provided, in section 4 the tool itself is shown and some of the design decisions that were taken during the process of creating the extension are outlined, and in section 5 first evaluation results are shown.

2. Relevance

Enterprise models are essential for the understanding and (re)design of an enterprise and its supporting software [2, 3] and can comprise many aspects [4, 5]. There is a growing need in the DEMO and Enterprise Modeling communities to have a tool that supports the creation of DEMO models that can be integrated with other models. Existing tools³ include Simplified Modeling Platform⁴, OpenModeling⁵, and DEMO world⁶, next to more ‘traditional’ drawing tools such as PowerPoint, Visio, and draw.io⁷. The latter drawing tools do not possess (semantic) knowledge about the concepts being modeled, and thus cannot assist in checking completeness or correctness of a model. The other tools, although there is some support to link DEMO to, e.g., ArchiMate models, seem to have the limitation that they are separated from existing tools for enterprise architecture modeling. It is considered relevant to have a tool for DEMO modeling that is available within an existing environment, so that users do not need to learn a new tool, and so that DEMO models can easily be connected with existing enterprise architecture models. Criteria in the choice of platform were:

1. The DEMO models can be integrated with concepts from other modeling languages, such as BPMN, UML, and ArchiMate;
2. The tool can incorporate scripts to automatically verify consistency and internal validity of the model;
3. Tables and (other) overviews can automatically be generated from the model; and
4. The tool is widely used.

Sparx EA is a platform to model enterprise architectures. It is widely used⁸ and recognized by Gartner since 2006⁹. Moreover, it allows for integration of several modeling languages and provides possibilities to create extensions (MDG: Model Driven Generation, also known as ‘Extension’) to facilitate other modeling languages, to add consistency checks, and to generate several reports from the model. As Sparx EA satisfies the criteria above, it was decided to create an extension to Sparx EA to facilitate the modeling of DEMO, with the potential to integrate it with other enterprise architecture models. Everyone that uses Sparx EA, can easily install the extension by importing it into Sparx EA, therefore allowing for wide adoption.

3. Background: DEMO modeling

DEMO (Design and Engineering Methodology for Organizations) is a leading method within the discipline of enterprise engineering [6], with strong methodological and theoretical roots [7, 8, 9, 10] that sets communication as the primal notion for the design of enterprises and its supporting

³<https://ee-institute.org/demo/tooling/>

⁴<https://simplified.engineering/>

⁵<https://openmodeling.ydns.eu/om226/index.htm>

⁶<https://www.demoworld.nl/Portal/Home>

⁷<https://www.drawio.com/>

⁸380.000 users (see <https://www.eausergroup.com/home>) and 720 companies (see <https://enlyft.com/tech/products/sparx-systems/>).

⁹<https://sparxsystems.com/press/articles/gartner.html>

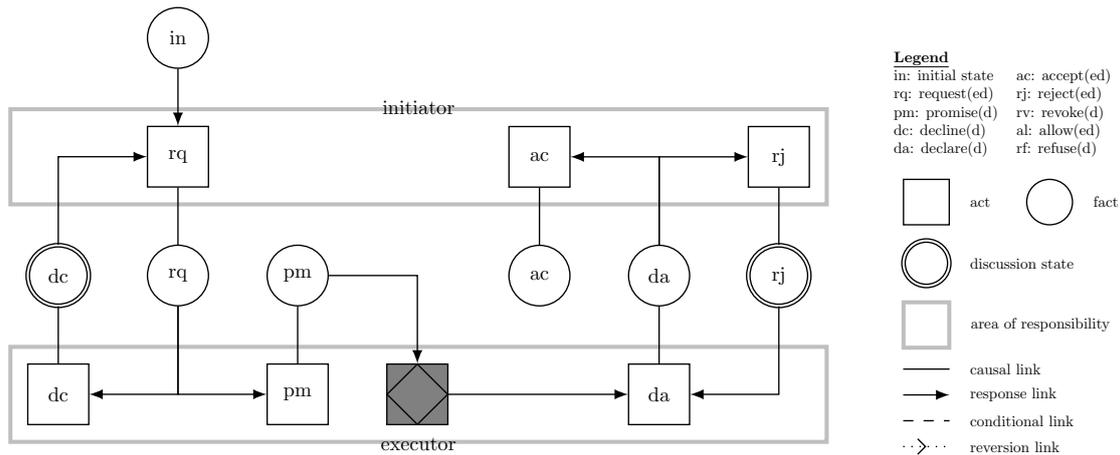


Figure 2: The standard transaction pattern shows the basic flow and discussion states. The complete transaction pattern includes four revocation patterns [1].

software systems [11]. The main principle behind DEMO modeling is that an enterprise is seen as a network of actors that enter into and comply with commitments about some product [1]. Such commitments are raised by actors in acts, the atomic units of action, and follow a generic pattern called the Complete Transaction Pattern (CTP). This CTP (see Figure 2) consists of 19 step kinds and deals with the basic flow – request, promise, execute, declare and accept – as well as discussion states – decline, reject – and cancellations (or revocations). The general working principle is that actors constantly check whether there are acts they have to deal with or respond to. The total set of acts for an actor to deal with is called the actor's *agenda*. By abstracting actors to *actor roles* and commitments regarding a specific *product kind* to *transaction kinds*, the model becomes independent of specific technologies, including human beings, that play a role in the operation. Figure 3 shows the 'compressed' CTP as one initiating actor role, one executing (responsible) actor role and a transaction kind (with related product kind).

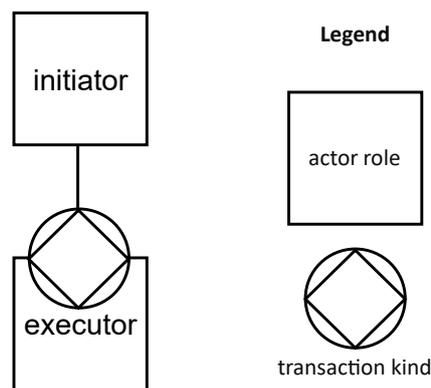


Figure 3: Transaction kind with initiating and executing actor roles; the CTP is 'hidden' in the transaction kind.

The DEMO ontological model of an enterprise consists of an integrated whole of four aspect models (see Figure 1), expressed in several diagrams and tables (see Figure 4) [1].

- The *Cooperation Model (CM)* models the *cooperation* of the enterprise; it consists of transaction kinds, associated (initiating and executing) actor roles, fact banks, access links between actor roles and fact banks, and wait links between transaction kinds and actor roles. The CM is expressed in one or more Coordination Structure Diagrams (CSDs) (see Figure 3) and a Transactor Product Table (TPT).
- The *Process Model (PM)* models the *processes* that take place as the effect of acts by actors, by detailing the coordination between actor roles; it makes explicit the causal and wait links between acts from the CTP. The PM is expressed in one or more Process Structure Diagrams (PSDs) and one or more Transaction Pattern Diagrams (TPDs).
- The *Fact Model (FM)* is the semantic model of *products* of the enterprise; it defines (declared or derived) fact types (entity types with their related product kinds, property types, attribute types and value types), existence laws and occurrence laws. The FM is expressed in an Object Fact Diagram (OFD) and zero or more Derived Fact Specifications (DFSs).
- The *Action Model (AM)* is the model of the *operation* of the enterprise, guiding actors in performing their acts. It specifies for every agendum kind with which the enterprise has to deal one or more Action Rule Specifications (ARSs). Each ARS fully supports the actor cycle and consists of an *event part* detailing the agendum kind to respond to, an *assess part* detailing the conditions to check and a *response part* that states how the actor should respond. Although these guidelines may look like rules, they offer the actors the possibility to autonomously – but responsibly – deviate from these guidelines.

The contents of the models are defined in the DEMO Specification Language (DEMO-SL) [12].

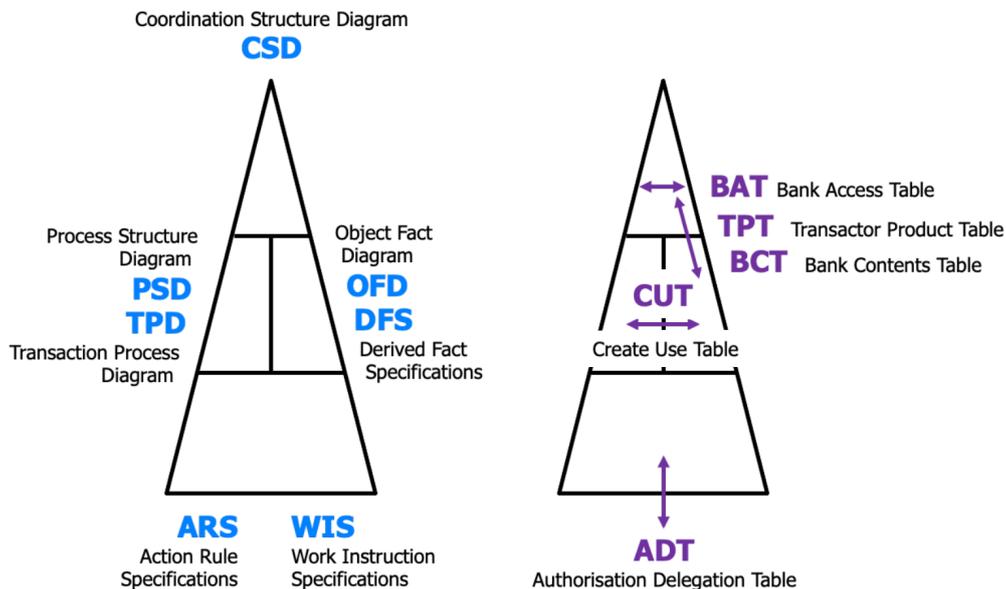


Figure 4: The ways of expressing the DEMO aspect models: diagrams (left) and tables (right) [1].

4. Tool: An extension (MDG) for Sparx EA

The created extension (MDG for Sparx EA) can easily be downloaded¹⁰; an installation and usage manual is included in the download. An Sparx EA installation (v16.1) is required for a correct working of the MDG – in earlier versions of EA the MDG will work, but layout may be different. The extension includes several scripts to renumber transaction kinds, to generate ARSs, and to reorder (clean up) the project directory.

Below, several screen shots are included to show how a DEMO model is created using the extension. The model represents the real-world Social Housing case [13], a Dutch governmental case about the registration for people with minimal income in order to obtain access to a low-rent house. Figure 5 shows the CSD; Figure 6 shows the TPT; Figure 9 shows the auxiliary diagram for TAR01 – this diagram will be explained in more detail below; Figure 7 shows the PSD; Figure 10 shows the OFD; and Figure 12, Figure 14 and Figure 16 show a selection of ARSs for TAR01. Example models will be published soon.

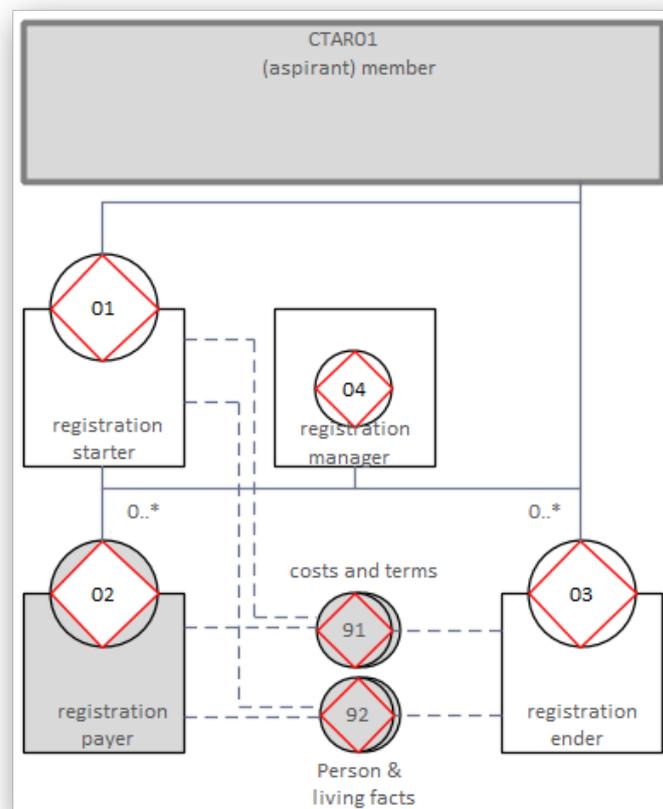


Figure 5: CSD for Social Housing, modeled in Sparx EA DEMO MDG

¹⁰<https://eexpertise.nl/cmsform.aspx?webpage=demo4mdg>

Transaction Product Table (TPT)					
transaction kind	product kind		executor role		
TK01	registration starting	PK01	[registration] is started	AR01	registration starter
TK02	registration paying	PK02	[registration] for [period] is paid	AR02	registration payer
TK03	registration ending	PK03	[registration] is ended	AR03	registration ender
TK04	registration management	PK04	registration management for [period] is done	AR04	registration manager

Figure 6: TPT for Social Housing, modeled in Sparx EA DEMO MDG

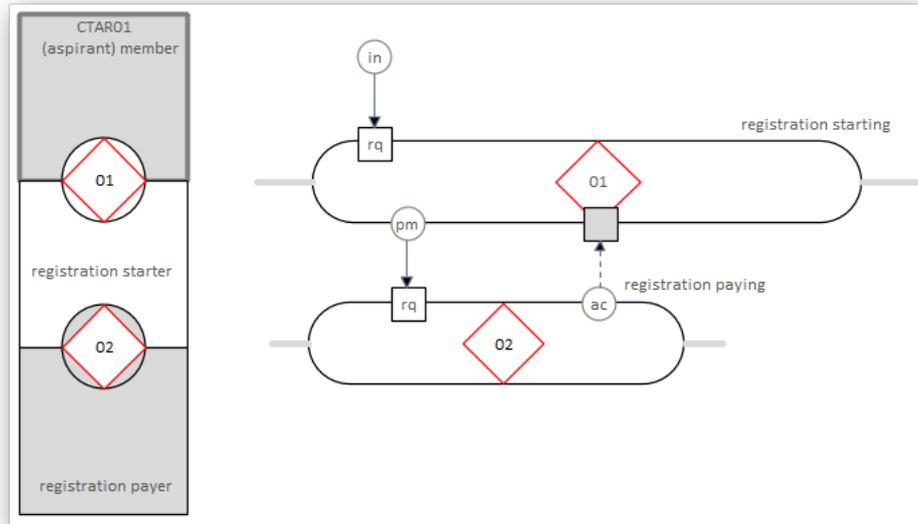


Figure 7: PSD for Social Housing, modeled in Sparx EA DEMO MDG

In order to ease the process of modeling a consistent set of DEMO models, several design decisions were taken that are outlined below.

4.1. Mandatory wait links for transaction composition

Sometimes, an actor role responsible for some transaction needs to initiate some other transaction, e.g., when a payment is needed before starting a registration. This is also known as transaction composition [14]. The DEMO way of thinking states that in that case the initiating actor role (usually) needs to wait for some state of the initiated transaction before proceeding with the transaction for which the actor role is responsible. This is denoted by a wait link from the initiated transaction kind to the initiating transaction kind. Figure 8 shows two of the possible variants of how two transaction kinds can be composed. Other (typical) variants include a causal link from T01/pm to T02/rq and a wait link from T02/pm or T02/ac to T01/ex.

While it is not made explicit in DEMO-SL, for this tool it is considered mandatory to have a wait link along with a causal link for transaction composition. The reasoning to do so follows from the way of thinking outlined above; If waiting for the initiated transaction is not needed,

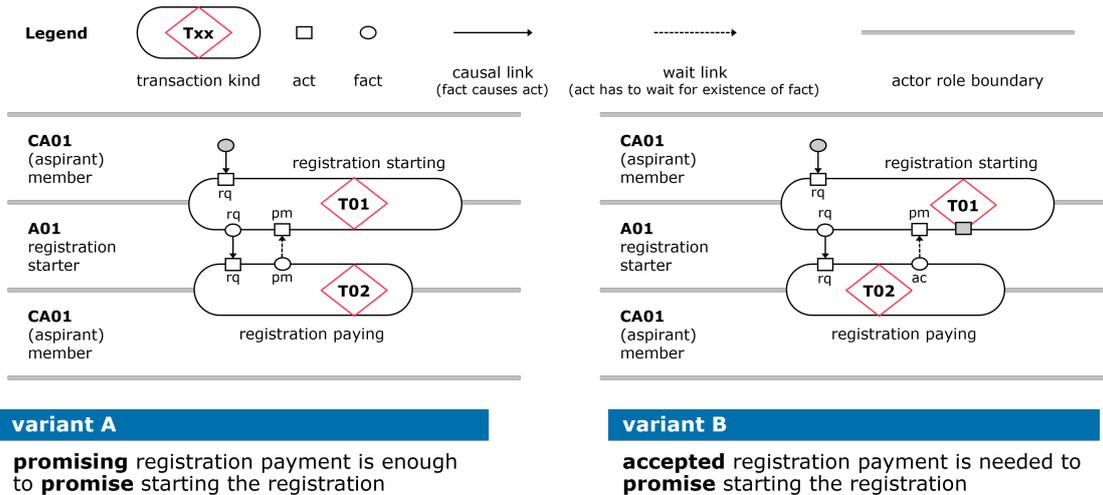


Figure 8: Two variants of transaction composition

why did the initiation even take place? Examination (see section 5) of several cases did not reveal a violation of this assumption.

4.2. Auxiliary diagram

The DEMO Way of Working prescribes to first create the CSD – perhaps for only a limited scope – at first – and then create the OFD. There is however quiet some overlap in these diagrams, as the product kinds are present in both diagrams, and also in the TPT and BCT (see Figure 4).

To support the modeler, an auxiliary diagram is introduced for each transaction kind. In such a diagram (see Figure 9), the transaction kind, along with its initiator role(s) and executor role,

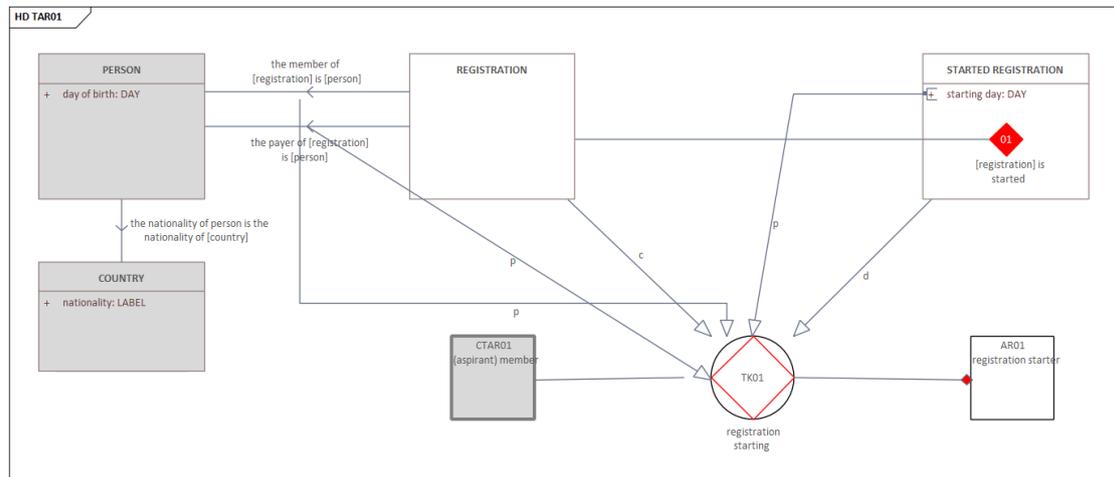


Figure 9: Auxiliary diagram for SH TAR01, modeled in Sparx EA DEMO MDG

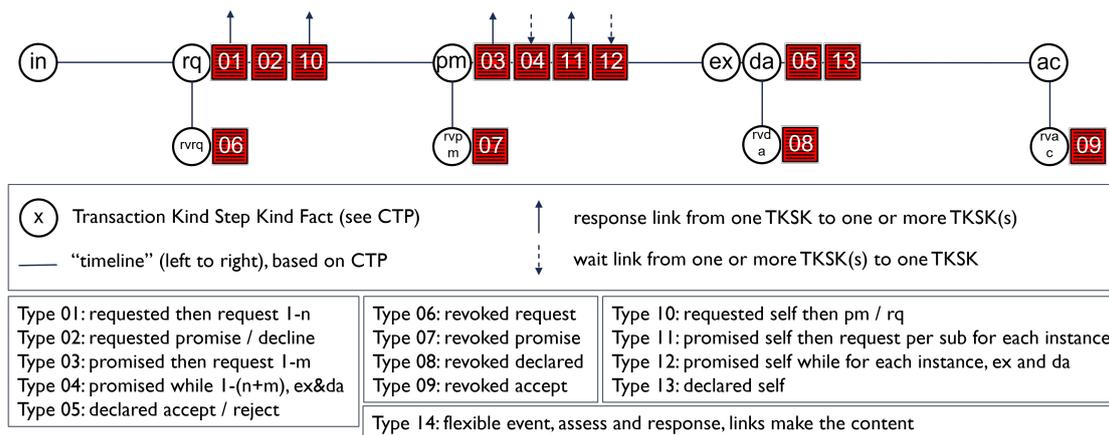


Figure 11: 14 ARS templates grouped on a timeline

DEMO-SL [12], is easily generated from the model.

4.3.1. Example: ARS01.2 (type 02)

Figure 12 shows ARS01.2, executed by TAR01. It is a ‘simple’ action rule that acts on the TK01/rq with possible responses being the TK01/pm and TK01/dc (type 02; see Figure 13 for the template). The auxiliary diagram (Figure 9) shows that REGISTRATION is the variable (case kind) of this transaction kind – it is the only entity type in this diagram with a c- or p-link to the transaction kind – and thus REGISTRATION is used as variable for the action rule. The auxiliary diagram also shows that the starting day and two properties of the registration are provided as parameter, and, as a result, these are automatically shown in the event part of the action rule. The truth clause of the assess part can then be modeled as a set of existence laws or business rules. The response part is automatically filled with the possible responses.

4.3.2. Example: ARS01.3 (type 03)

Figure 14 shows ARS01.3, executed by TAR01. As shown in the PM (Figure 7), in this ARS TK02 needs to be requested, which conforms to type 03 (see Figure 15). Visually it is shown which transaction kind is requested from the response part, including the necessary and created fact kinds.

4.3.3. Example: ARS01.4 (type 04)

Figure 16 shows ARS01.4, executed by TAR01. As shown in the PM (Figure 7), this ARS needs to wait for the TK02/ac before proceeding with TK01. It therefore conforms to type 04 (see Figure 17). Visually it is shown which transaction kind and act kind needs to be waited for.

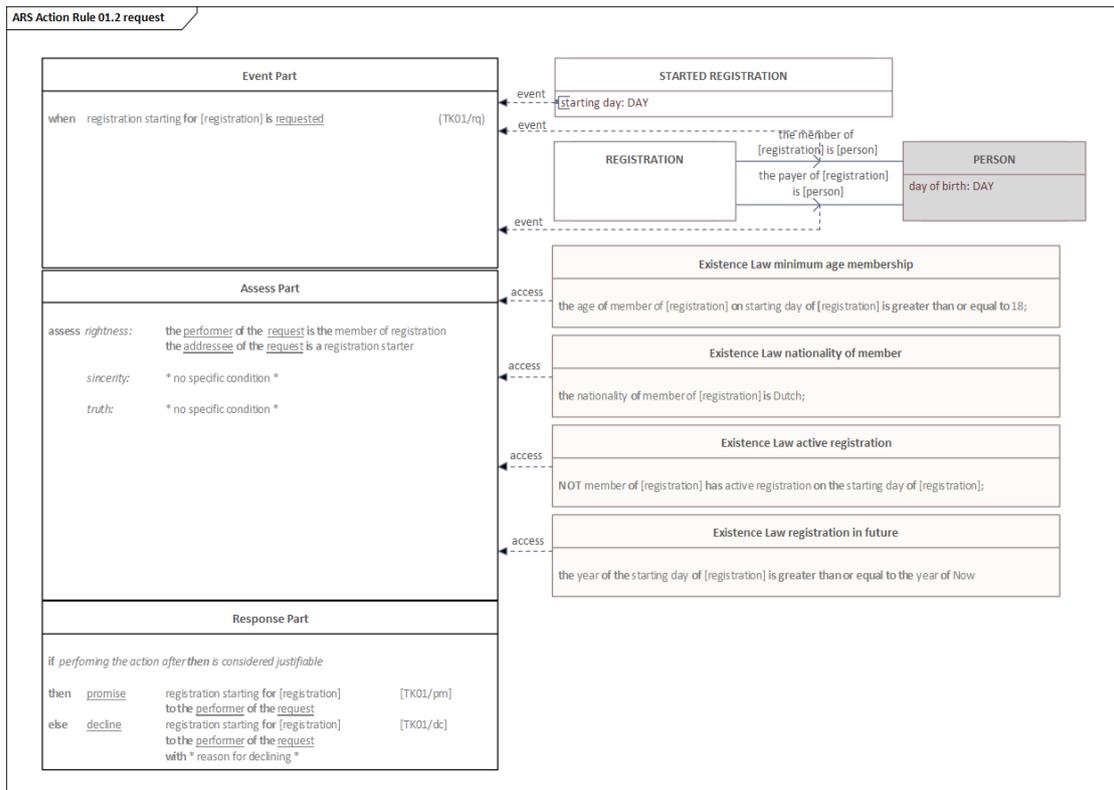


Figure 12: ARS01.2 (TK01/rq) for Social Housing, modeled in Sparx EA DEMO MDG

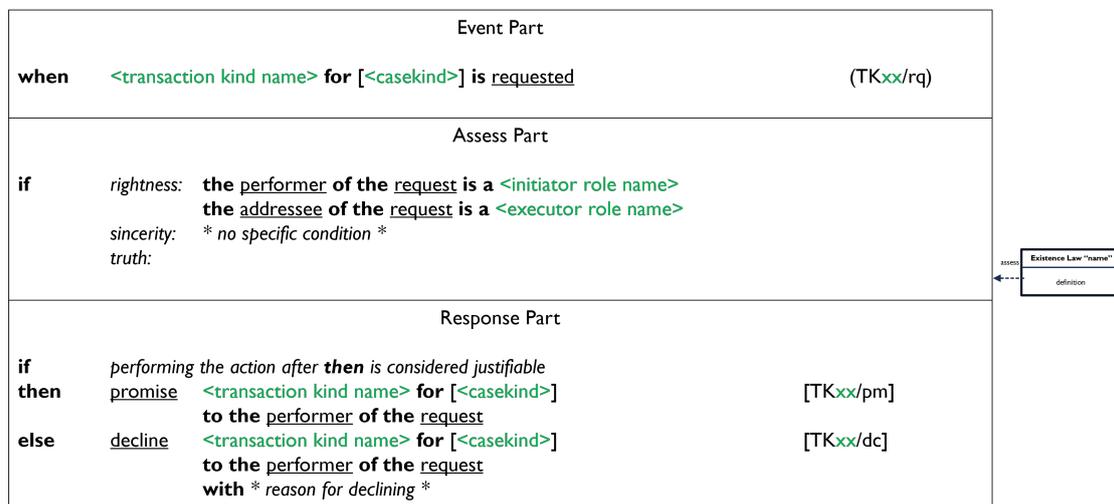


Figure 13: ARS template type 02; in green and between chevrons (<>) are the parameters with which this template is converted to an actual ARS

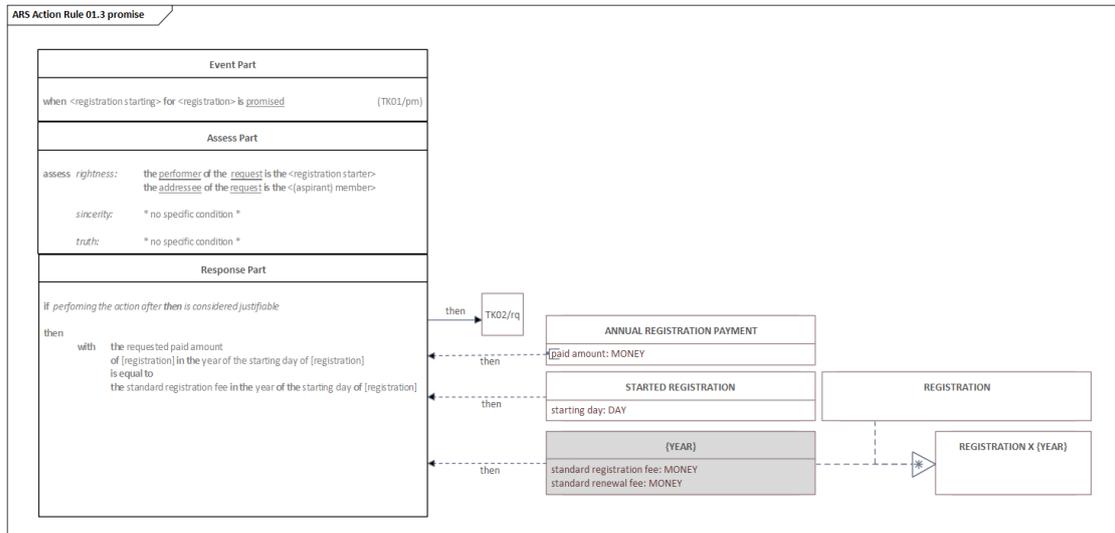


Figure 14: ARS01.3 (TK01/pm) for Social Housing, modeled in Sparx EA DEMO MDG

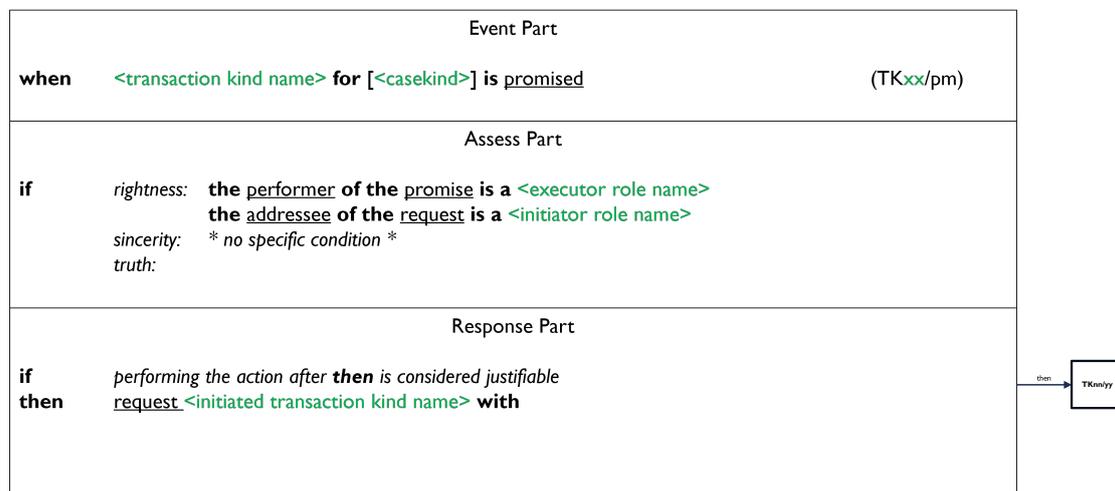


Figure 15: ARS template type 03

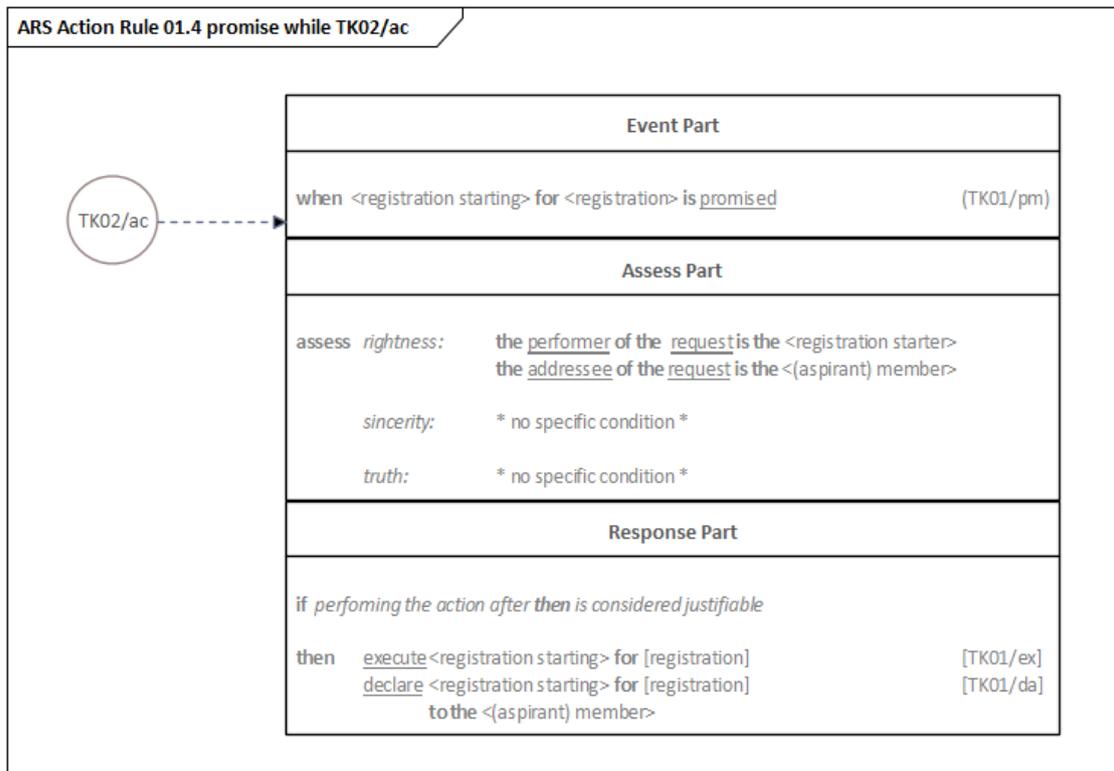


Figure 16: ARS01.4 (TK01/pm while TK02/ac) for Social Housing, modeled in Sparx EA DEMO MDG

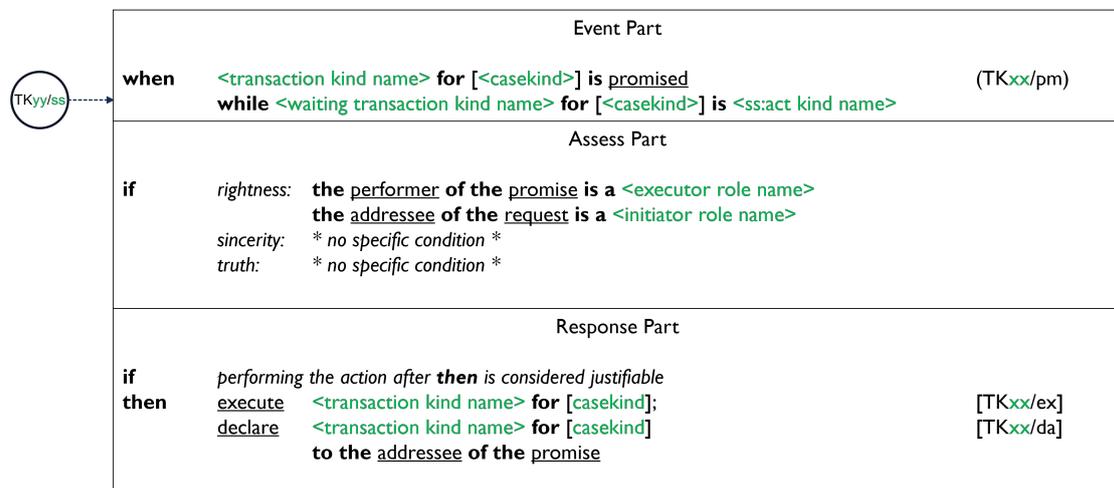


Figure 17: ARS template type 04

5. Evaluation

The extension was tested with several cases, including the more academic case Desperados [15] and the real-world case Social Housing (shown above). Modeling these cases with the tool greatly enhanced and improved it in several iterations. It also showed that the fourteen templates provide enough flexibility to model all cases and are helpful for modelers to easily decide on the patterns for the action rules. Currently, multiple people from multiple enterprises are using the tool, including Rijkswaterstaat, the executive agency of the Dutch Ministry of Infrastructure and Water Management. Evaluations include:

- “The Sparx EA DEMO MDG is a valuable modeling tool DEMO that users have been waiting for. As it is available as an MDG for Sparx EA, users can use all the features of Sparx EA, including a repository for storing the model, easy drop-and-drag functionality, managing the properties of elements, and scripts to automate tasks. The DEMO MDG has several clever features that increase the efficiency of creating and maintaining DEMO models while at the same time it keeps the models consistent.” (Jeroen Graave)
- “The MDG v1.3 (beta) consists of several scripts that accelerate the creation of DEMO models; it generates several diagrams based upon other diagrams. I especially like the DEMO clean-up kit that helps organising the repository. The tooling used to automatically generate Transaction Process Diagrams is amazing and takes away a lot of repeated work. Templates are delivered for all ARS types that can be accompanied with links for while, with, response, conditional and existence laws. This beta version still has some annoying minor bugs but I have seen this tool being developed from version 1.2 to 1.3 and with every new version the number of bugs is declining.” (anonymous business architect)

In future, we plan for more elaborate, both quantitative and qualitative, evaluations. The roadmap includes more automation in creating tables and diagrams, extended possibilities to connect DEMO concepts to concepts from ArchiMate and other modeling languages, and generation of working software from these models.

Acknowledgments

We want to thank Han van Roosmalen from EExpertise for his endless efforts in creating the scripts that support this Sparx EA extension. We want to thank the reviewers of PoEM and EDEWC 2023 for their feedback that led to several improvements of this paper.

References

- [1] J. L. G. Dietz, J. B. F. Mulder, *Enterprise Ontology – A Human-Centric Approach to Understanding the Essence of Organisation*, The Enterprise Engineering Series, Springer, Cham, 2020. doi:10.1007/978-3-030-38854-6.
- [2] M. S. Fox, M. Gruninger, *Enterprise Modeling*, *AI Magazine* 19 (1998). doi:10.1609/aimag.v19i3.1399.

- [3] F. Vernadat, Enterprise modelling: Research review and outlook, *Computers in Industry* 122 (2020). doi:10.1016/j.compind.2020.103265.
- [4] U. Frank, Multi-perspective enterprise modeling: foundational concepts, prospects and future research challenges, *Software and Systems Modeling* 13 (2014) 941–962. doi:10.1007/s10270-012-0273-9.
- [5] K. Sandkuhl, J. Stirna, A. Persson, M. Wißotzki, *Enterprise Modeling: Tackling Business Challenges with the 4EM Method*, Springer, 2014. doi:10.1007/978-3-662-43725-4.
- [6] J. L. G. Dietz, J. A. P. Hoogervorst, A. Albani, D. Aveiro, E. Babkin, J. Barjis, A. Caetano, P. Huysmans, J. Iijima, S. van Kervel, H. Mulder, M. Op 't Land, H. A. Proper, J. Sanz, L. Terlouw, J. Tribolet, J. Verelst, R. Winter, The discipline of enterprise engineering, *International Journal of Organisational Design and Engineering* 3 (2013) 86–114. doi:10.1504/IJODE.2013.053669.
- [7] J. L. Austin, *How to do things with words*, William James Lectures, Oxford University Press, 1962.
- [8] J. R. Searle, *Speech Acts: An Essay in the Philosophy of Language*, Cambridge University Press, Cambridge, London, 1969.
- [9] J. Habermas, *The theory of communicative action*, Cambridge: Polity Press, 1986.
- [10] H. Weigand, Two decades of language/action perspective, *Natural Language Engineering* 49 (2006) 45–46.
- [11] V. E. van Reijswoud, J. B. F. Mulder, J. L. G. Dietz, Communicative Action Based Business Process and Information Modelling with DEMO, *The Information Systems Journal* 9 (1999) 117–138.
- [12] J. L. G. Dietz, *The DEMO Specification Language v4.7*, Technical Report, Enterprise Engineering Institute, 2022. URL: <https://ee-institute.org/download/demo-specification-language-4-7-1/>.
- [13] M. Op 't Land, M. R. Krouwel, S. Gort, Testing the Concept of the RUN-Time Adaptive Enterprise - Combining Organization and IT Agnostic Enterprise Models with Organization Implementation Variables and Low Code Technology, in: D. Aveiro, G. Guizzardi, R. Pergl, H. A. Proper (Eds.), *EEWC 2020: Advances in Enterprise Engineering XIV*, volume 411 of *Lecture Notes in Business Information Processing*, Springer, 2021, pp. 228–242. doi:10.1007/978-3-030-74196-9_13.
- [14] S. Guerreiro, P. Sousa, A framework to semantify BPMN models using DEMO business transaction pattern, *CoRR* (2020). doi:10.48550/arXiv.2012.09557.
- [15] J. Dietz, E. van Dipten, M. Krouwel, P. Kuipers, T. de Mik, I. Theuwissen-Krol, *Begrijpen en Maken van Essentiële Modellen, Werken met DEMO (WmD)*, Enterprise Engineering Institute, 2021. URL: <https://ee-institute.org/demo/werken-met-demo/>, dutch.