Empirical evaluation of a new DEMO modelling tool that facilitates model transformations

Thomas Gray[1], Marné De Vries[2[0000-0002-1715-0430]]

[1, 2] University of Pretoria, Department of Industrial and Systems Engineering, South Africa
Thomas.Gray@up.ac.za; Marne.DeVries@up.ac.za

Abstract. The *engineering methodology for organizations* (DEMO) incorporates an *organization construction diagram* (OCD) and *transaction product table* (TPT) to depict a consolidated representation of the enterprise in terms of *actor roles* that coordinate in consistent patterns on different *transaction kinds*. Although managers find the OCD useful due to its high level of abstraction, enterprise implementers and operators prefer detailed flow-chart-like models to guide their operations, such as *business process model and notation* (BPMN) models. BPMN models are prevalent in industry and offer modeling flexibility, but the models are often incomplete, since they are not derived from theoretically-based, consistent coordination patterns. This study addresses the need to develop a DEMO modeling tool that incorporates the novel feature of transforming user-selected parts of a validated OCD, consistently and in a semi-automated way, into BPMN collaboration diagrams. The contribution of this article is two-fold: (1) to demonstrate the *utility* of the new DEMO-ADOxx modelling tool, including its model transformation ability; and (2) to empirically evaluate the *usability* of the tool.

Keywords: DEMO, BPMN, ADOxx, model transformation, software modelling, multi-view modelling.

1. Introduction

Domain-specific languages are created to provide insight and understanding within a particular domain context and stakeholder group. As an example, the *design and engineering methodology for organizations* (DEMO) provides models that represent the organization domain of an enterprise [1]. DEMO offers a unique design perspective, since its *four aspect models* represent *organization design domain* knowledge in a concise and consistent way, removing technological realization and implementation details [1]. One of DEMO's aspect models, the construction model, incorporates an *organization construction diagram* (OCD) that provides a concise representation of *enterprise operation*. Managers find the OCD useful due to its high level of abstraction. Yet, enterprise implementers and operators prefer detailed flow-chart-like models to guide their operations, such as *business process model and notation* (BPMN) models. BPMN models are prevalent in industry and offer modeling flexibility, but the models are often incomplete, since they are not derived from theoretically-based, consistent coordination patterns [2]. Others [3] identified the need to generate BPMN models from

DEMO models, based on transformation specifications. Yet, the specifications did not consider the complexity of hierarchical structures in DEMO models. In addition, their transformation specifications were not supported by tooling to automate DEMO-BPMN transformations [4].

A new DEMO-ADOxx tool, called DMT, addresses the need to compile a DEMO construction model, in accordance with the specifications stated in [5] and [6]. In addition, the tool incorporates the novel feature of transforming user-selected parts of a validated OCD, consistently and in a semi-automated way, into BPMN collaboration diagrams [4].

This article has two main objectives: (1) demonstrating a main feature of the new DEMO-ADOxx tool, i.e. transforming OCD parts into BPMN collaboration diagrams for a complex scenario; and (2) empirically evaluating the usability of the tool.

The article is structured as follows. Section 2 provides background on DEMO models and the development of the DMT. Section 3 suggests a research method to evaluate the DMT, whereas section 4 presents the evaluation results, concluding in section 5 with future research directions.

2. Background

In this section we provide background theory on DEMO, present a demonstration case with sufficient complexity to validate the DMT against specifications for DMT in the form of: (1) a meta-model for the OCD and TPT, and (2) OCD-BPMN transformation specifications.

DEMO uses four linguistically based *aspect models* to represent the ontological model of the *organisation domain* of the enterprise, namely the *construction model* (CM), *process model* (PM), *action model* (AM), and *fact model* (FM) that exclude technology implementation details [5]. A qualitative analysis on DEMO aspect models, indicate that the CM, detailed by the PM, is useful for assigning responsibilities and duties to individuals [7]. According to a study by Decosse et al. [7], the AM and FM *"are necessary if you are going to develop or select applications"*. The conceptual knowledge embedded in the PM is similar to the BPMN collaboration diagram [3]. Yet, BPMN is widely adopted by industry [8] and facilitates simulation and workflow automation, as demonstrated by BPMN-based industrial tools, such as ADONIS, Camunda and Bizagi. Our initial DEMO-ADOxx tool thus focused on representing the CM. Rather than using a PM as the next step of modelling, our tool incorporates a user-interface to capture parent-part structures from the modeler that would normally be indicated on a PM, bridging the gap from the CM to detailed and consistent BPMN diagrams.

The CM is expressed using three diagrams: (1) the *organisation construction diagram* (OCD); (2) the *transaction product table* (TPT) and (3) the *bank contents table* (BCT). We incorporated specifications regarding the OCD and TPT, as stated in [5] and [6], as well as BPMN 2.0 [9] for the first version of the DMT. The *way of modelling* in [10], indicates that a modeller has to validate the definition of each transaction kind (TK) by defining an associated product kind (PK). Due to their tight coupling, the OCD and TPT were incorporated in the first version of our tool. We excluded the BCT, since the "BCT can only be completed when the FM is produced" [1, p 272].

2.1. The Demonstration Case

The demonstration case had to be of such complexity that a modeler would be able to construct a TPT (a list of TKs and PKs - not shown here due to space restrictions) and an OCD (illustrated in Fig. 1). The case represents the *universe of discourse and some operations at a fictitious college*. In accordance with the guidelines presented in [10], our demonstrating OCD, portrayed in Fig. 1, only includes TKs that are of the *original transaction sort*. **Bold** style indicates the type of construct whereas *italics* refers to an instance of the construct (see Fig. 1).

Scope of Interest (SoI) indicates that the modeler analyses a particular scope of operations, namely *some operations at a college*. Given the SoI, Fig. 1 indicates that three **environmental actor roles** are defined, see the grey-shaded constructs *student*, *project sponsor* and *HR of project sponsor* that form part of the environment. Within the SoI, multiple **transaction kinds (TKs)** are linked to different types of **actor roles** via **initiation links** or **executor links**. As an example, *supervisor allocation* (*T01)* is a **TK** that is initiated (via an **initiation link**) by the **environmental actor role** *student (CA01)*. In accordance with [10], the *student (CA01)* is by default also regarded to be a **composite actor role** "of which one does not know (or want to know) the details". Since *T01* is linked to an **environmental actor role**, it is also called a **border transaction kind**. *T01* is executed (via the **executor link**) by the **elementary actor role** named *supervisor allocator (A01)*.

All the other actor roles in Fig. 1 within the **SoI** are **elementary actor roles**, since each of them is only responsible for executing one **transaction kind**. A special case is where an **elementary actor role** is both the **initiator** and **executor** of a **transaction kind**, also called a **self-activating actor role**. Fig. 1 exemplifies the **self-activating actor role** with *module reviser (A04)* and *project controller (A05)*. Since **actor roles** need to use facts created and stored in transaction banks, an **information link** is used to indicate access to facts. As an example, Fig. 1 indicates that *project controller (A05)* has an **information link** to **transaction kind** *module revision (T04)*, indicating that the *project controller (A05)* uses facts in the transaction bank of *module revision (T04)*. It is also possible that **actor roles** within the **SoI** need to use facts that are created via **transaction kinds** that are outside the **SoI**. As an example, Fig. 1 indicates that **actor roles** within the **SoI** (called, *some operations at a college*) need to use facts that are created outside the SoI and stored in the transaction banks of **aggregate transaction kinds**, namely *person facts* of *AT01*, *college facts* of *AT02*, *accreditation facts* of *AT03*, *timetable facts* of *AT04* and *student enrollment facts* of *AT05*. According to Fig. 1, the *student enrollment facts* of **aggregate transaction kind** *AT05* are not accessed by any **actor roles**, which should be possible (according to the meta-model depicted in [5]).

Even though Fig. 1 only includes **elementary actor roles** within the **SoI**, it is possible to consolidate **elementary actor roles** within a **composite actor role**, where a composite actor role "is a network of transaction kinds and (elementary) actor roles" [10]. Fig. 1 illustrates two **composite actor roles** within the **SoI,** namely *College (CA0)* and *Controller (CA01)*. Both *CA00* and *CA01* encapsulate a number of transaction kinds and elementary actor roles.

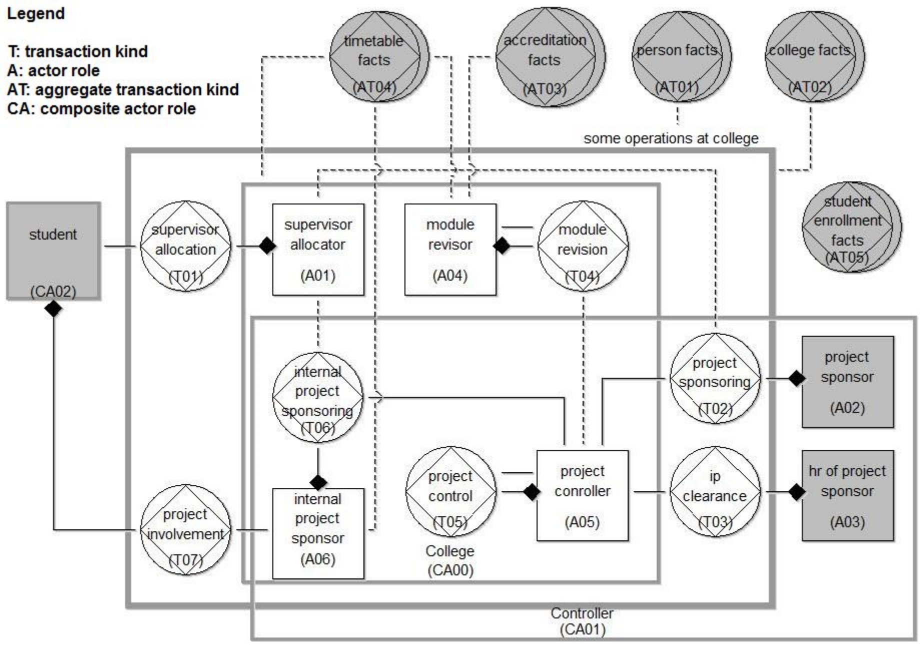


Fig. 1. The OCD for a college, based on [5]

2.2. General DEMO Tool Requirements, specifications and the new DMT

Previous work [4] highlighted five *minimum requirements* from a tertiary educational perspective for a DEMO modelling tool and compared existing tools against these requirements. The requirements indicated that a DEMO modelling tool should be *comprehensive*, supporting all four DEMO aspect models (R1). The tool should support the most *recent language specification* and facilitate future upgrades of the DEMO language (R2). The tool should facilitate *model transformations* to other modelling languages, such as BPMN (R3). The tool should be available at *low cost* for educational purposes (R4), and it should be *usable* (R5).

A comparison of existing tools indicated that existing tools do not fulfil the minimum requirements. None of the tools, except the new DMT, facilitated model transformations [4]. The main objective of the new DMT was to demonstrate the *transformation feature*, but also allow for future development of the tool. Even though the DMT addressed the minimum requirements and initial usability tests on DMT were positive [4], additional evaluation was needed, especially in terms of its *utility* (R2 and R3) and *usability* (R5). In terms of R2, the DMT had to address the existence rules encapsulated in the meta-model of the OCD and TPT (see [4]). For R3, the DMT had to address all *four transformation scenarios*, depicted in [11]. This article demonstrates the most complex scenario of the four, where one **TK** has multiple parts, i.e. the **actor role** that **executes** the user-selected **TK**, is also **initiating** one or more other **TKs**.

Referring to Fig. 1 the TK labelled *T05* (*project control*) is executed by **actor role** *A05* (*project controller*). The same **actor role** *A05* (*project controller*) also **initiates** multiple other **TKs**, namely *T02* (*project sponsoring*), *T03* (*IP clearance*), and *T06* (*internal project sponsoring*). The DMT was realized as an OMiLAB project which enables free download: https://austria.omilab.org/psm/content/demo [4].

3. Research Method

According to Bagozzi [12] the *unified theory of acceptance and use of technology* (UTAUT) exemplifies the complexity of technology assessment with UTAUT including multiple variables for predicting intentions and predicting behaviour. For this article, our aim was not to perform a comprehensive assessment of DMT, but to prioritise two key variables to guide our decision-making in pursuing further development of the DMT. Empirical evaluation of two critical variables were needed: *utility* and *usability.*

The *utility tests* were performed internally, categorized as *existence rules* tests and *transformation* tests. The *existence rules* tests had to ensure that the DMT facilitated creation of a sound OCD and TPT in accordance with the meta-model. We used the demonstration case presented in section 2.1 to compile valid tests. As an example, one of the existence rules state: *1..* TK is contained in CAR 0..**, meaning that *a transaction kind is contained in zero-to-many composite actor roles*, and *a composite actor role contains one-to-many transaction kinds*. Using the case study presented in Fig. 1, two test cases were compiled: (1) It should be possible to create T01 that is not contained within the CAR CA00 (College) and (2) It should also be possible to create T06 as a part within CAR CA00 (College). The *transformation tests* tested the tool's ability to semi-automate four main OCD-BPMN transformations, associated with the demonstration case presented in section 2.1.

The DMT was empirically validated from a user perspective for *usability,* i.e. using a survey-based approach to measure the DMT's ease of *modelling and validating the OCD and TPT*; and *generating a BPMN diagram*, based on a user-selected TK on the OCD. Survey modelling methods are often used to gather opinions, impressions and beliefs of human subjects via questionnaires [13]. Organisations and individuals will only adopt a new tool if the perceived usefulness is high. We have used a standardized questionnaire due to several advantages stated by [14]: objectivity, replicability, quantification, economy, communication and scientific generalization. The SUMI (Software Usability Measurement Inventory) questionnaire, developed by the Human Factors Research Group (HFRG) at the University College Cork, is widely used and cited in national an international standards with a global reliability of 92% [14]. The 50-item questionnaire has five subscales for efficiency, affect, helpfulness, control, and learnability, with a three-point Likert scale to rate each item as *disagree*, *undecided* and *agree*. The results of the five subscales are consolidated into a global scale that provides an indication of the software's general usability [14]. The questionnaire would be useful to evaluate the *overall usability* of the DMT for its current scope.

A laboratory experiment (in accordance with [15]) was conducted with 34 post graduate students as participants. The participants all attended a post-graduate module on enterprise design. One of the module components incorporated training on DEMO. Participants also received training on the demonstration case depicted in section 2.1 before they modelled the OCD and TPT, using the DMT, also experimenting with the OCD-BPMN transformation feature. Afterwards, participants had to complete the SUMI questionnaire, availed via the SUMI online form [16]. We also documented all comments or questions posed by the participants during the experiment.

4. Evaluation Results

For the *first utility evaluation* objective we tested whether the DMT-created models comply with the *existence rules* that are specified for the OCD and TPT between the following concepts: (1) aggregate transaction kind (ATK); (2) composite actor role (CAR); (3) elementary actor role (EAR); (4) fact kind (FK); (5) independent P-fact kind (IFK); and (6) transaction kind (TK) (see [4]). Table 1 presents the *existence rule* (grey shaded), *test cases* and *test results*. We only elaborate on the *test results* if additional explanation is needed. The results indicate that the DMT addressed all the existence rules that apply to the OCD and TPT.

Table 1: Existence rules, test cases and results

From TK: 1..1 the product kind of TK is IFK 1..1
Forward: When T01 is defined, it should be linked to one product kind. When viewing the TPT, the single product kind should be displayed for the TK. **Reverse:** When defining the product kind P02, it should be for one TK. **Forward Results:** The Model Analysis/Validation feature highlights production kinds with no product kind defined. Thus mandatory 1..1 is not enforced. **Reverse Results:** The product kind ID is automatically generated and not editable. The system also blocks any attempt to duplicate product kind names.
From TK: 1..* TK is contained in ATK 0..*
Forward: *Outside SoI:* For DEMOSL 3.7 it should not be possible to indicate the parent-part relationship. *Inside SoI:* It should be possible to create T01 in accordance with Fig. 1 where T01 is not contained in an ATK. It should be possible to create T02, where T02 is contained in the ATK T05. T05 is an ATK with multiple parts (i.e. T02, T03, T06, T07). **Reverse:** *Outside SoI:* For DEMOSL 3.7 it should not be possible to indicate the parent-part relationship. *Inside SoI:* It should not be possible to initially model T05 as an ATK, grey-shading T05, without indicating the parts, i.e. without modelling the parts T02, T03, T06, T07). The software should only allow T05 as a composite (without parts) if created outside the SoI. **Reverse Results:** An ATK placed within the SOI is colored red to indicate non-validity. The Model Analysis/Validation feature also highlights it as an error.
From TK: 1..* TK is contained in CAR 0..*
Forward: It should be possible to create T01 that is not contained within the CAR CA00 (College). It should also be possible to create T06 as a part within CAR CA00 (College). **Reverse Results:** It should be possible to create CAR CA00 (College) with multiple TKs, namely T04, T05 and T06.
From EAR: 1..* EAR is an initiator role of TK 0..*
Forward: It should be possible to create A01 as per Fig. 1 where A01 is not initiating other TKs. It should be possible to create A05, initiating T02, T03 and T06.

Reverse: It should be possible to create T06, initiated by A05 as per Fig. 1. In addition, it should also be possible to create T06, also initiated by A01 (i.e. A01 is also initiating T06), even though this scenario is not evident on Fig. 1.
From EAR: 1..1 EAR is the executor role of TK 1..1
Forward: When A01 is modelled as an elementary actor role without any execution link attached, a validation message should be shown when validating the model. When A01 is modelled as an elementary actor role, it should not be possible to connect both A1 as the executor for T01 and A01 as the executor for another TK, say T08 (supervision). Note that T08 is not displayed on Fig. 1, but has to be created temporarily to do this test. **Reverse:** When T01 is modelled as a TK without any execution link attached, a validation message should be shown when validating the model. When T01 is linked via an execution link to A01 and T01 is linked to another EAR, say A08 (supervisor), an error message should appear to indicate that a TK may only have one executor. **Reverse Results:** A warning message is displayed and the second connection is not possible.
From EAR: 0..* EAR may inspect the contents of bank TK 0..*
Forward: When A01 is created without any inspection links, no validation rules should appear when saving the model. It should also be possible to link A01 to T02 via an inspection link, as well as A01 to T06, with no validation errors appearing when the model is saved. **Reverse:** It should be possible to create the TK T03 with no inspection links attached. It should also be possible to create an inspection link between T02 and A01, as well as an inspection link from T02 to A06.
From EAR: 0..* EAR is contained in CAR 0..*
Forward: It should be possible to create an actor role, say A08 (supervisor) within the SoI, but outside the CAR CA00 (College), without displaying validation errors when saving the model. It should be possible to create A06 as an EAR embedded in the CAR CA00 (College), also embedded in the CAR CA01 (Controller). **Reverse:** It should be possible to create the CAR CA00 (College) without embedding any TKs, linking it via an execution link to T01. It should be possible to create the CAR CA00 (College) with multiple embedded EARs, i.e. A01, A04, A05 and A06.
From EAR: 0..* EAR has access to the bank of ATK 0..*
Forward: It should be possible to create the EAR A1 with no inspection links to AT01 and AT02, i.e. no validation errors on saving. It should also be possible to create EAR A4 with inspection links to AT03 and AT04 without validation errors on saving. **Reverse:** It should be possible to create AT05 without any inspection links attached, with no validation errors on saving. It should be possible to create both an inspection link from AT04 to A04, as well as an inspection link from AT04 to A06, without validation errors on saving.
From CAR: CAR is a specialization of EAR
The relations of the EAR should also be available for the CAR [6]. Hence, when creating CAR CA00 (College) without any embedded detail, it should be possible to link the CA00 via an execution link to T01, link CA0 via an initiation link to T07, link CA00 via an initiation link to T02, link CA00 via an initiation link to T03, and link CA00 via an inspection link to AT04.
From CAR: CAR is a part of CAR
As explained by [6], the SoI is a special case of a CAR. It should be possible to create a CAR, i.e. CA00 (College) within the SoI without any validation errors when saving the model.

For the *second utility evaluation* objective we tested whether a DMT model, generated via DMT's transformation feature, could address the complexity of a TK that has multiple parts. When a modeler selects the TK labelled *T05* (*project control*) on the OCD presented in Fig. 1, a BPMN collaboration diagram is generated, presented in

Fig. 2. Addressing threats to validity on the transformation-abilities of DMT, we have also used a second demonstration case, i.e. the Rent-a-car case from [10] to validate the transformation abilities in terms of all *four transformation scenarios*. Due to space limitations, we could only include one demonstration case (in section 2.1) in this paper. The second demonstration case highlighted shortcomings in the transformation specifications of [11], indicating that the four transformation scenarios had to be extended further.

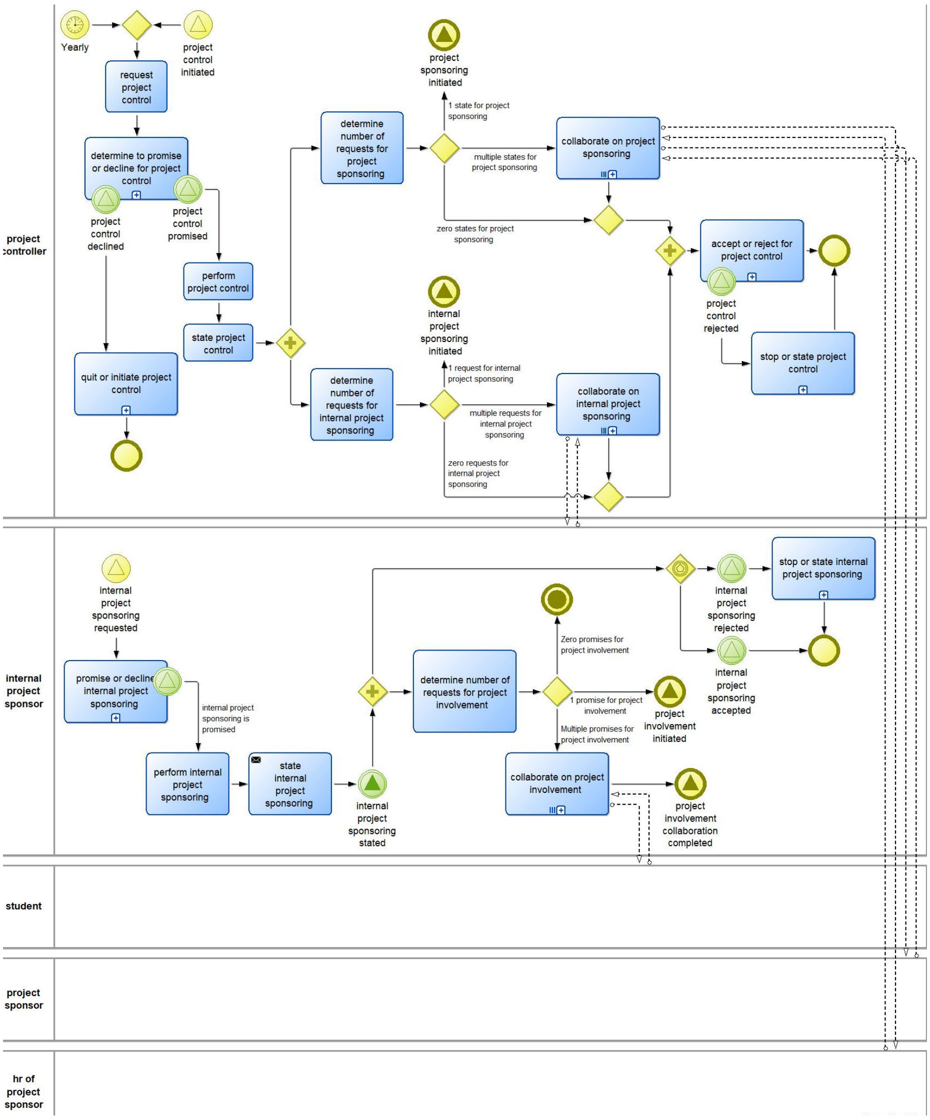
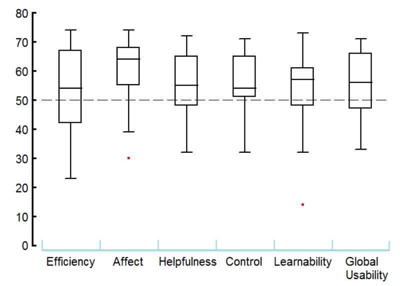


Fig. 2. A BPMN collaboration diagram, generated from T05 (project control)

The *usability results* are summarised in Table 2. The results draw a very positive picture, especially considering the prototypical nature of the DMT. The tool is evaluated positive in all five sub-categories and in the global scale. The highest value was found in the category *affect* which measures the participants emotional feeling mentally stimulated and pleasant, or the opposite: stressed and frustrated as a result of interacting with the tool. The results indicate that 31 out of 34 participants perceived the DMT as being *important* or *extremely important* for supporting their task. Most noted things to improve: *Link Usage (7)*, *Menu (3)*, *General Usage (5)*, *Error Handling (4)*. Nine participants did not mention any necessary improvements. Most noted positive things: Ease of use (15), Intuitivity (7), Model Transformation (7), Interface (6), and Drag & Drop (4).

Table 2. Results of the SUMI questionnaire

	Global	**Efficiency**	**Affect**	**Helpfulness**	**Controllability**	**Learnability**
Mean	56.5	54.0	60.8	54.9	55.4	54.6
Std. Dev.	10.8	13.5	10.6	10.9	10.3	11.4
Median	56.5	54.5	64.5	55.0	54.5	57.0



Based on the qualitative feedback on their questionnaire and interactive feedback during the experiment, we have already incorporated the suggestions within the DMT.

5. Conclusions and Future Research Directions

The main objective of this paper was to empirically evaluate the new DMT in terms of utility and usability. The *utility tests* were performed internally as follows: (1) The *existence rules* associated with the meta-model for the OCD and TPT, were translated into test cases, based on the demonstration case presented in section 2.1; and (2) *Conformance to OCD-BPMN transformation specifications*, tested the tool's ability to semi-automate four main transformations, of which be presented the most complex transformation in Fig. 2. For *usability*, we used the SUMI questionnaire to evaluate the ease-of-use of the DMT.

Our evaluation results are positive regarding both *utility* and *usability*. Since we empirically evaluated the DEMO tool within a laboratory setting, future work is envisaged to evaluate the tool within a real-world enterprise setting. Sauro & Lewis [14] also suggest comparative usability tests, e.g. comparing the new DMT with another existing DEMO modelling tool.

The DMT was developed and evaluated, based on DEMOSL 3.7 (see [5]) as well as extensions (see [6]). Yet, a newer version, DEMOSL 4.5 has been published recently (see [1]). Modelling languages evolve and enforce the evolution of associated models. Realizing the tool as an open source project, using the ADOxx platform within the OMiLAB digital ecosystem, ensures that not only the authors, but also the OMiLAB community, can contribute towards future tool enhancements [17].

Acknowledgements: We would like to thank Dominik Bork for his continued support and collaboration on the DMT project.

References

1 Dietz, J. L. G., Mulder, H., B, F.: Enterprise ontology: A human-centric approach to understanding the essence of organisation. Springer International Publishing AG (2020)
2 Caetano, A., Assis, A., Tribolet, J.: Using DEMO to analyse the consistency of business process models. In: Moller, C., Chaudhry, S. (eds.) Advances in Enterprise Information Systems II. pp. 133-146. Taylor & Francis Group, London (2012). DOI: 10.1201/b12295-17
3 Mraz, O., Náplava P, Pergl R, Skotnica, M.: Converting DEMO PSI transaction pattern into BPMN: A complete method. In: Aveiro, D., Pergl, R., Guizzardi, G., Almeida, J. P., Magalhães, R., Lekkerkerk, H. (eds.) LNBIP 284. pp. 85-98. Springer International Publishing, Cham, Switzerland (2017). DOI: 10.1007/978-3-319-57955-9_7
4 Gray, T., Bork, D., De Vries, M.: A new DEMO modelling tool that facilitates model transformations. In: Nurcan, S., Reinhardt-Berger, I., Soffer, P., Zdravkovic, J. (eds.) Enterprise, Business-Process and Information Systems Modeling. Vol. LNBIP 387, pp. 359-374. Springer Nature Switzerland, Cham, Switzerland (2020). DOI: 978-3-030-49418-6_25
5 Dietz, J. L. G., Mulder, M. A. T.: DEMOSL-3: DEMO specification language version 3.7. SAPIO (2017)
6 Mulder, M. A. T.: Towards a complete metamodel for DEMO CM. In: al., D. e. (ed.) OTM 2018 Workshops, LNCS 11231. pp. 97-106. (2019). doi.org/10.1007/978-3-030-11683-5_10
7 Decosse, C., Molnar, W. A., Proper, H. A.: What does DEMO do? A qualitative analysis about DEMO in practice: Founders, modellers and beneficiearies. In: Aveiro, D., Tribolet, J., Gouveia, D. (eds.) Advances in Enterprise Engineering VIII. Springer, Portugal (2014). doi.org/10.1007/978-3-319-06505-2_2
8 Grigorova, K., Mironov, K.: Comparison of business process modeling standards. Int. J. of Eng. Sci. & Manag. Res., **1**(3), 1-8 (2014)
9 Object Management Group: Business process model & notation, https://www.omg.org/bpmn/, last accessed 2019/30 May
10 Perinforma, A. P. C.: The essence of organisation (3rd ed.). Sapio, www.sapio.nl (2017)
11 De Vries, M., Bork, D.: Bridging organization design knowledge and executable business processes: A model transformation approach based on DEMO and BPMN. (In Review)
12 Bagozzi, R.: The legacy of the technology acceptance model and a proposal for a paradigm shift. J. of the Assoc. for Inf. Syst., **8**(4), (2007). DOI: 10.17705/1jais.00122
13 Siau, K., Rossi, M.: Evaluation techniques for systems analysis and design modelling methods – a review and comparative analysis. Inf. Syst. J., **21**(3), 249-268 (2011). doi.org/10.1111/j.1365-2575.2007.00255.x
14 Sauro, J., Lewis, J. R.: Quantifying the user experience: practical statistics for user research. Elsevier Inc., Waltham, USA (2012)
15 Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., Wesslén, A.: Experimentation in software engineering. Springer, Heidelberg (2012)
16 HFRG: Software Usability Measurement Inventory, http://sumi.uxp.ie/en/index.php, last accessed 2020/17 January
17 Bork, D., Buchmann, R. A., Karagiannis, D., Lee, M., Miron, E.-T.: An open platform for modeling method conceptualisation: The OMiLAB digital ecosystem. Commun. of the AIS, **44**(32), 673-697 (2019). doi.org/10.17705/1CAIS.04432