J. Michael, J. Pfeiffer, A. Wortmann (Hrsg.): Modellierung 2022 Satellite Events,
Digital Library, Gesellschaft für Informatik e.V. 245

MEMO4ADO: A Comprehensive Environment for Multi-Perspective Enterprise Modeling

Alexander Bock[1], Ulrich Frank[1], Monika Kaczmarek-Heß[1]

Abstract: This paper describes the construction and use of a comprehensive tool for enterprise modeling, MEMO4ADO, implemented using the meta modeling environment ADOxx. MEMO4ADO offers an extensible set of editors for various modeling languages. The paper gives an overview of the tool's architecture and describes the integration of modeling languages and diagrams. A use case serves to illustrate the design and analysis of enterprise models with MEMO4ADO. The paper concludes with a brief assessment and an outlook on the further development of the tool.

Keywords: meta modeling; DSML; enterprise modeling tool; MEMO; MEMO4ADO; ADOxx

1 Introduction

The ability of companies to compete in present-day markets depends increasingly on the effectiveness of their information systems (IS). To exploit the potential of information technology (IT), it is usually necessary to reorganize organizational processes and structures alongside the introduction of new software systems. The recognition of this interdependence of business and IT led more than three decades ago to the emergence of enterprise modeling [Za87]. An enterprise model integrates models of *information systems*, such as data models or object models, with models of a company's *action system*, like models of business processes or of organizational structures. In the decades following the publication of Zachman's pioneering, but relatively coarse high-level framework in the 1980s [Za87], a variety of more sophisticated methods for enterprise modeling have been developed [FS98, Sa14], among them *Multi-Perspective Enterprise Modeling* (MEMO) [Fr14].

MEMO is an extensive method for enterprise modeling. It supports various perspectives on the enterprise, which can be described in more detail using an extensible set of domain-specific modeling languages (DSMLs). The set of DSMLs includes a language for modeling organizational structures and business processes (OrgML) [Fr11a, Fr11b], as well as languages for describing goal systems (GoalML) [OFK15], decision processes (DecisionML) [Bo15], and IT infrastructures (ITML) [FKHdK21]. In addition, MEMO integrates general-purpose modeling languages such the ERM and UML class diagrams.

The evolution of MEMO has been accompanied by the development of modeling tools from the very beginning. Earlier modeling tools for MEMO were implemented on the

[1] Universität Duisburg-Essen, [alexander.bock,ulrich.frank,monika.kaczmarek-hess]@uni-due.de

basis of Smalltalk [Fr94] and the EMF [GF10]. *MEMO4ADO* is the latest in this series of tools. It is implemented using the meta modeling environment ADOxx [FK13], with the first version having been presented in 2015 [BF16]. Since then, MEMO4ADO has been intensively used for teaching purposes, and it has been extended and revised on a regular basis [FB20, FKHdK21]. MEMO4ADO is freely available at `https://austria.omilab.org/psm/content/memo4ado/info`.

This paper provides an overview of the tool's foundation, its latest version, and demonstrates how to use it. To this aim, first, an overview on the MEMO's language architecture and its implementation in ADOxx is provided. Then, the implemented languages, their integration as well as model editors are shortly described. Finally, a use case pointing to the way the MEMO4ADO tool may be used to analyse or design an enterprise model is provided. In opposition to already published papers on MEMO4ADO, e.g., [BF16, FB20], we focus here on the tool itself and functionalities it offers, as well as newly added languages. The paper concludes with final remarks on the tool usage and next steps.

2 MEMO's Language Architecture and its Implementation in ADOxx

As already mentioned, MEMO4ADO is implemented on the basis of the meta modeling environment ADOxx [FK13]. As indicated in the introduction, we had used EMF for a previous implementation of a MEMO modeling environment, and struggled with the effort required to implement a new DSML and to maintain the tool. In contrast, ADOxx turned out to be better suited to satisfy our requirements. In particular, it offers the ability to efficiently and effectively implement new DSMLs and corresponding modeling editors. The possibility to easily create stand-alone modeling tools has also influenced our choice.

Fig. 1 illustrates the language architecture of MEMO and the implementation in ADOxx. As is seen on the left-hand side, the various MEMO modeling languages are specified with a common meta modeling language, the MEMO MML [Fr11c]. The different languages are integrated through common concepts, such that an integrated object model can be generated for the implementation of modeling tools. Adaptations to the original meta models are, consequently, propagated consistently to the unified object model of the modeling tool.

As is seen on the right-hand side of Fig. 1, the ADOxx environment enabled us to follow a largely analogous scheme in the practical implementation. Each MEMO meta model was reconstructed using the ADOxx meta modeling editor (called 'ADOxx Development Toolkit'). The common concepts that integrate the different MEMO languages were specified using the ADOxx's capacity of cross-diagram links, cf. *interref* [FK13, p. 7]. These links enable the modeler to define references from elements in one diagram (e.g., a business process diagram) to elements in another (e.g., an IT infrastructure diagram). Using this feature, the close integration of the MEMO language could be successfully carried over to the MEMO4ADO tool.

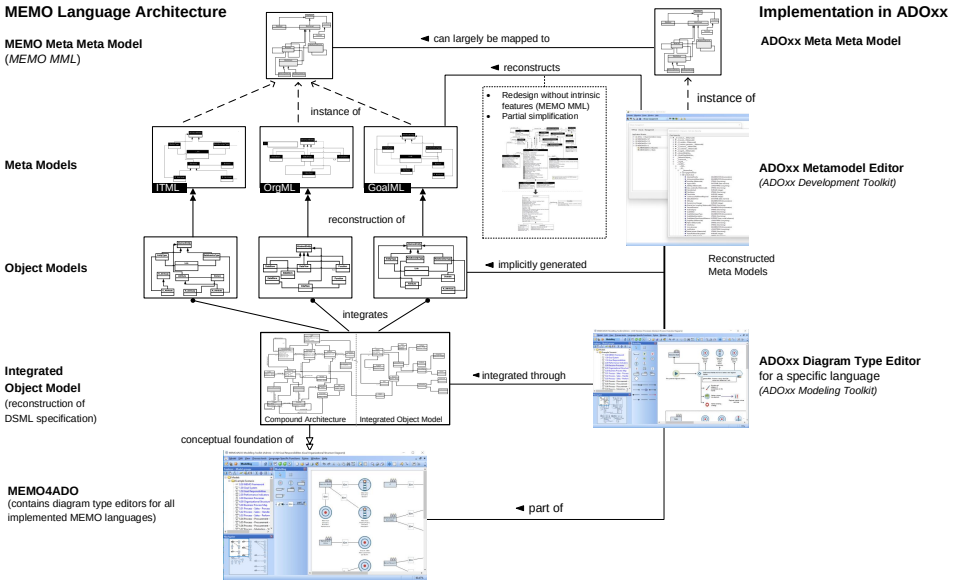

Fig. 1: Language architecture and mapping to tool architecture

Only two kinds of modifications were required during the reconstruction of the MEMO meta models in ADOxx. First, the MEMO MML [Fr11c] features so-called ‘intrinsic features’. These are used to mark metaclasses, attributes, or associations which are specified at level M2 but instantiated only at level M0. The modeling environment of ADOxx, however, is confined to level M1; it is not possible to instantiate elements again into instances at level M0. Accordingly, we had to partially redesign the MEMO languages without intrinsic attributes. The second group of modifications covers some minor simplifications in the service of accessibility. For example, the complex MEMO GoalML originally required separate definitions of goals and their objects, but it turned out to be more practical to reunite these aspects into one concept.

On the basis of the reconstructed meta models and certain other specifications in the ADOxx Development Toolkit (including the definition of the concrete syntax), the diagram editors for all implemented languages could be generated automatically. The resulting tool, MEMO4ADO, integrates all implemented MEMO languages in the form of different diagram types (e.g., ‘Goal System Diagrams’, ‘Organizational Structure Diagrams’, and others). These diagram types and their use will be sketched in the following sections. For a more comprehensive description of MEMO4ADO and its implementation, see [BF16, FB20].

3 Implemented Languages, Integrated Diagram Types and Features of Model Editors

The current version of MEMO4ADO (version 1.10) implements, among others, the MEMO GoalML [OFK15] to model organizational goal systems, the MEMO OrgML to model organizational structures [Fr11a], the MEMO OrgML to model business processes [Fr11b], the MEMO DecisionML [Bo15] to model organizational decision processes, the MEMO MetricML [St12] to model performance indicator systems, and the MEMO ITML to model IT infrastructures [FKHdK21]. Each of these DSMLs allows for the creation of dedicated diagrams of specific types. All DSMLs are integrated through a common meta model and through common concepts. Thus, elements of one model may reference elements of other models, cf. Sec. 2. This enables a targeted navigation through an enterprise model. While each DSML is supported by a specific model editor (e.g., 'Business Process Control Flow Diagram' created using the OrgML, bottom right in Fig. 2, or 'IT Infrastructure Diagram' created using the ITML, top right in Fig. 2), it is also possible to design diagrams that represent an integrated view on models created with different DSMLs, cf. [BF16, FB20]. For example, in the integrative diagram type 'Goal-Organizational Structure Diagram' it is possible to link goals defined in a 'Goal System Diagram' and organizational units defined in an 'Organizational Structure Diagram' (bottom left in Fig. 2).

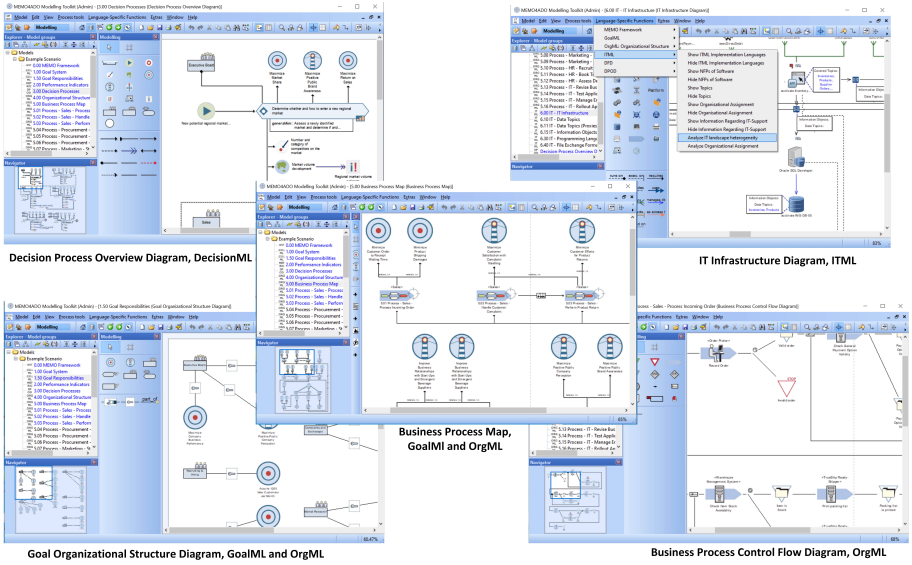

Fig. 2: MEMO4ADO Model Editor and Selected Diagram Types

All DSMLs implemented in MEMO4ADO offer rich sets of modeling concepts that allow for the design of sophisticated models. Let us exemplary consider an 'IT Infrastructure Diagram', cf. Fig. 3, being the essential MEMO ITML diagram type, used to design, re-design or analyze an organization's IT infrastructure. As such, this diagram type permits to describe

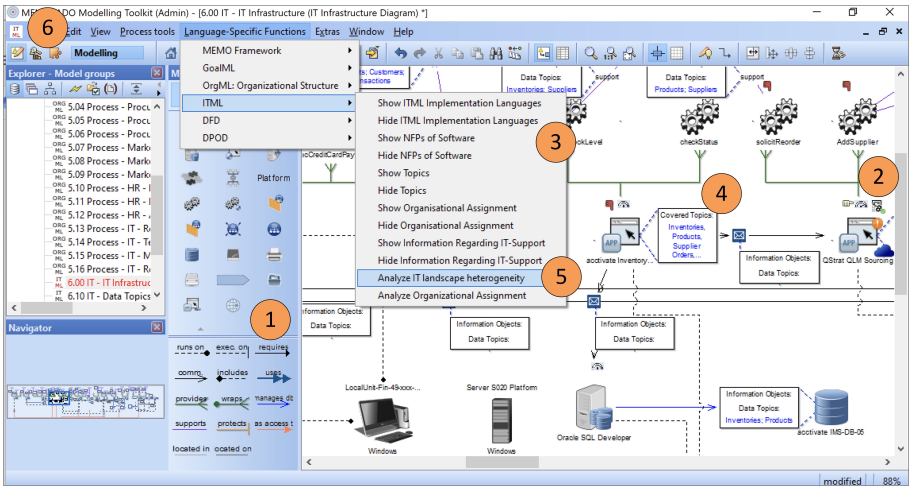

Fig. 3: MEMO4ADO: Exemplary IT Infrastructure Diagram

platforms (both physical as well as virtual ones), hardware artifacts, software systems, services, as well as various dependencies among those, cf. (1). Each kind of artifact comes with a variety of attributes. For many attributes, dedicated notational symbols are provided, whose appearance is automatically adjusted based on the current attribute value (e.g., mission critical, user satisfaction, runs in cloud, cf. (2)). A variety of relationships between modeling concepts, e.g., 'runs on', 'communicate', serve the differentiated description of dependencies between software and/or hardware artefacts.

In order to ease language use and to facilitate additional analyses for different target groups, further functions have been added. Examples of those are, cf. also [BF16, FB20]: (a) *Different Levels of Notational Details*, cf. Fig. 3, (3) – as some diagrams exhibit a level of notational detail which might be considered too complicated for some purposes, functions have been implemented that allow to switch between different levels of details. (b) *Overview Textboxes*, cf. Fig. 3, (4) – attribute values of model elements can be specified and investigated using the ADOxx notebook dialog. Sometimes, it may also be considered helpful to see values of attributes at a glance while interpreting a diagram. For this purpose, we implemented various additional text box views for each language that attach a basic text box to each model element in which the values for selected attributes are listed textually. (c) *Dedicated language-specific functions*, cf. Fig. 3, (5) – for selected languages, a various additional functions allowing to conduct specific kinds of analysis, e.g., analysis of heterogeneity of IT landscape[2], of organizational assignment, have been implemented. (d) *Ad-hoc Queries*, cf. Fig. 3, (6) – note that in addition to the above-mentioned features, the generic query

[2] Here the modeled elements are analysed to identify such aspects as, among others, types of platforms used, types of operating systems, programming languages used to implement software artifacts. Users are then presented with a corresponding table detailing the variety of those aspects and the number of elements identified.

mechanism offered by ADOxx may be also used to answer questions/conduct analyses of interest, also spanning multiple diagrams. Queries can be executed on models using the ADOxx Query Language (AQL) through the "Analysis" component. In short, the AQL query language allows to formulate queries on models in a style similar to SQL, cf. [FK13, p. 18], thus allowing to retrieve elements fulfilling some criteria, e.g., elements of some type, elements having some attribute's value, and/or related to some other element(s). The AQL queries can either be entered manually by a user or can be pre-defined, cf. [FK13, p. 18], to support specific analysis scenarios as required by the given modeling method being implemented. When accessing the pre-defined query, a user has then the option to select required elements from the lists showing all available elements matching the criteria defined in the query definition, and then execute the query.

4 Use Case: Analysis and Design of a Partial Enterprise Model

The basic way of using the MEMO4ADO tool comes down to (1) creating different core diagrams that describe selective abstractions of the enterprise in question (i.e., goal systems, organizational structures, business process control flows, and IT infrastructure), (2) to subsequently interrelate them, and finally (3) conducting integrative, reflective analysis on both action system and information system in tandem. Possible application scenarios range from strategic goal planning processes to integration of organisational IT landscape. For details, see [BF16, FB20].

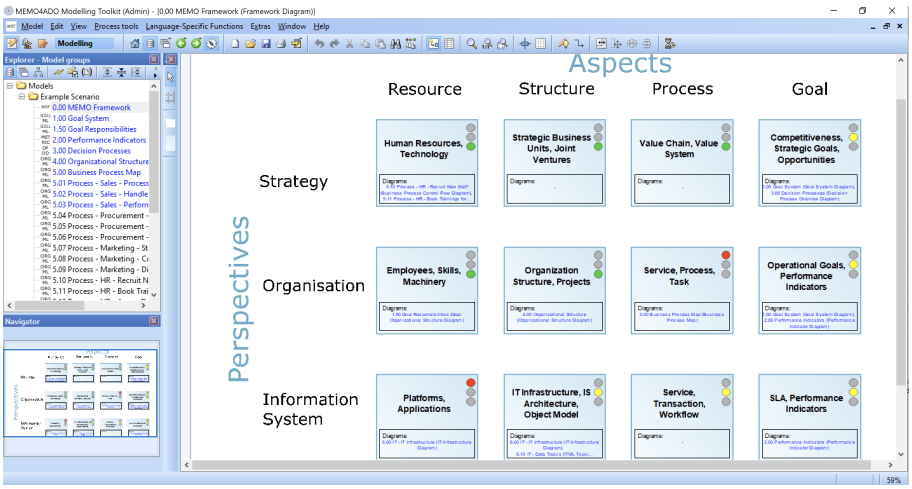

Fig. 4: MEMO4ADO: MEMO Generic Framework

The following example serves to demonstrate how the MEMO4ADO tool can be used to design and analyze an enterprise model. Due to the limited extent of the paper, the example is highly simplified. Let us assume that in order to stay competitive, a wholesale company that sells tools and material mainly to handicraft businesses has to analyze and eventually

redesign its services and processes. First, a common understanding of the company and the problems it is facing needs to be established. To that end, MEMO4ADO provides a generic framework that allows for a high-level overview of a company's current or future situation. For instance, the framework presented in Fig. 4 structures an enterprise along two dimensions. The first dimension encompasses three perspectives: (1) strategic, (2) organizational, and (3) information system. The second dimension encompasses different generic aspects found in each perspective, such as structure, process, and resources. The generic framework provides a a common starting point for identifying key problem areas and for defining priorities at the beginning of a project. The representation of the framework in a two-dimensional table allows assigning specific topics to each focus, and thus, not only provides a first overview of the enterprise and the current situation, but also allows to identify areas in need of a more elaborate analysis. To further investigate specific topics, diagrams can later be assigned to each focus. This way, the generic framework can be used as a central starting point to navigate an enterprise model, cf. Fig. 4.

The company for instance, may decide to increase its competitiveness by adapting its business model (e.g., by focusing on highly customizable solutions, as well as offering additional services to improve customer experience) with a specific focus on the automation of its processes. The design of a preliminary goal model (referenced from the MEMO framework) serves to analyze the implementation of such a strategy, cf. Fig. 5. The corresponding strategy may in turn require re-engineering (sales) processes. The efficient execution of these processes requires accounting for the supporting IT infrastructure as well as for the corresponding organizational structure. To this end, the company first designs models of targeted business processes, like detailing the 'Sales - Incoming Order Process' (see Fig. 7).

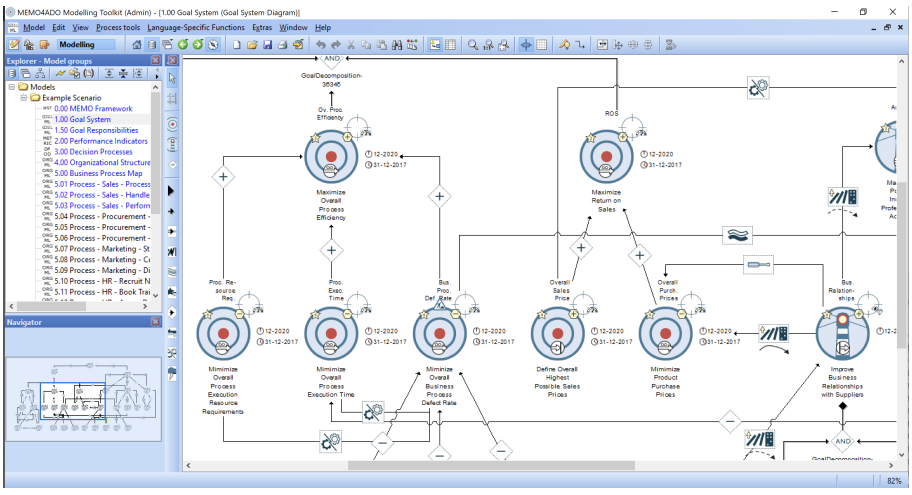

Fig. 5: MEMO4ADO: Goal System Diagram

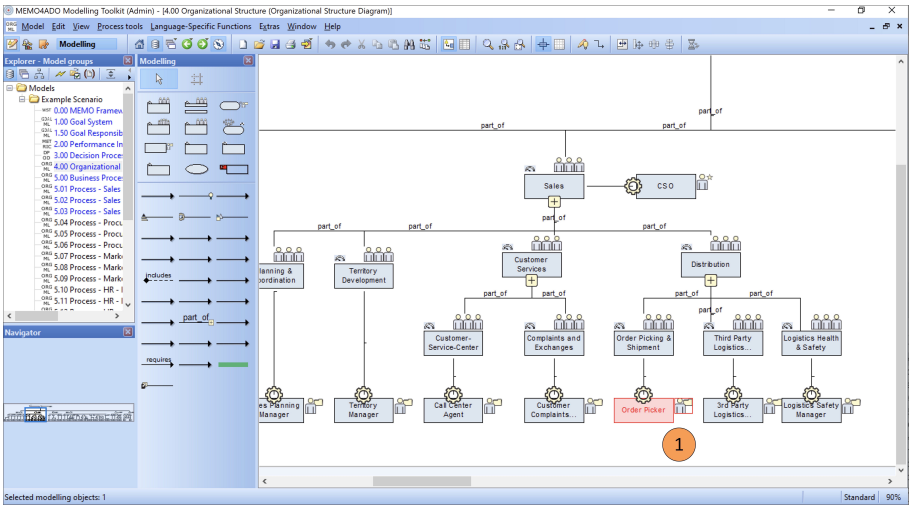

Fig. 6: MEMO4ADO: Organizational Structure Diagram

Those process models are used to guide the incremental development of corresponding models of the IT infrastructure and the organizational structure.

Thus, the process model in Fig. 7 does not only show an excerpt of a model that describes a customer order process, but it also indicates how a business process model is integrated with other models, cf. (1) and (2). On the one hand, links to services enable the integration

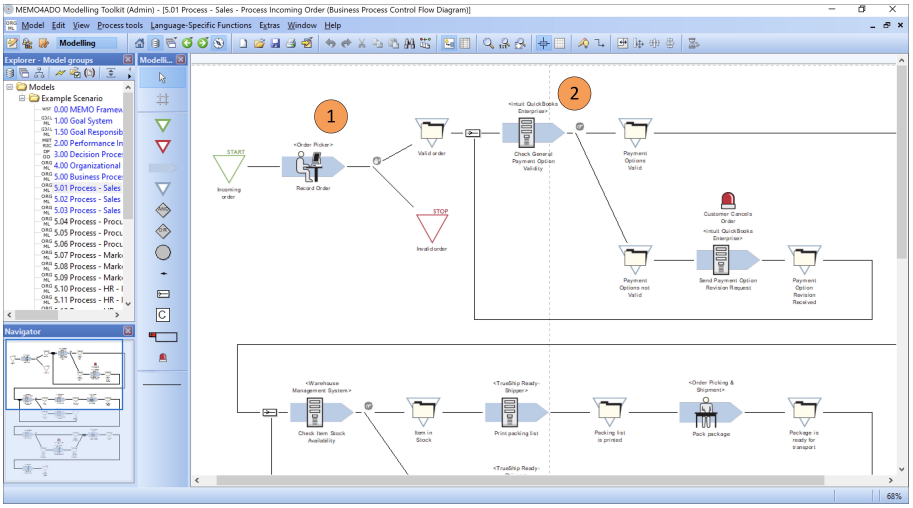

Fig. 7: MEMO4ADO: Business Process Control Flow

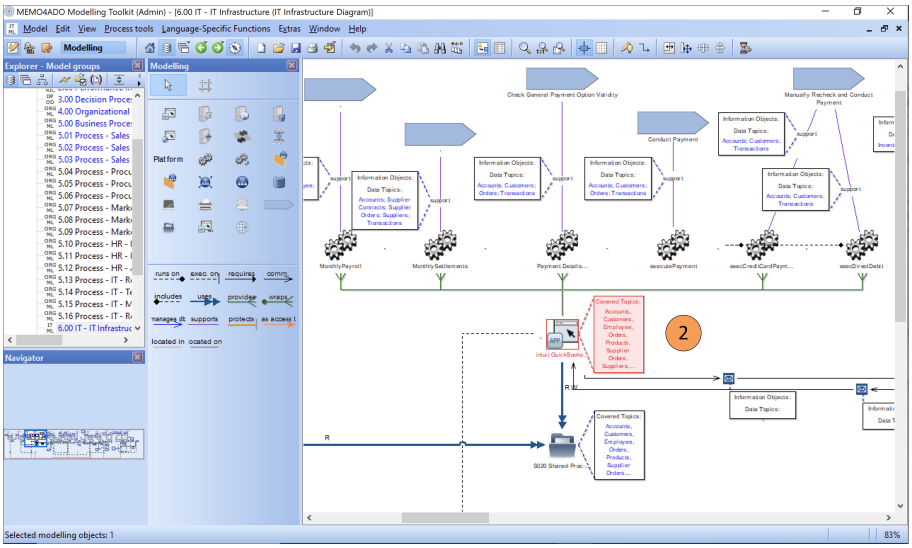

Fig. 8: MEMO4ADO: IT Infrastructure Diagram

of the business process model with a model of the IT infrastructure, cf. 'intuit QuickBooks Enterprise' Application Software, Fig. 8, supporting the activity 'Check General Payment Option Validity', Fig. 7, (2). On the other hand, the organizational units that are assigned to each activity in the process model serve as links to corresponding elements in a related model of the organization structure, cf. 'Order Picker', Fig. 6, (1) performing the activity 'Record Order', Fig. 7, (1).

The level of detail of models will often gradually increase as a modeling project progresses. Take, for example, the analysis of investments into IT required to support the prospective business model. In this case, it might be reasonable to first create a high-level model of the current IT infrastructure, in order to assess its potential to realize future IT services. In a next step, a more elaborate model of a revised IT infrastructure could be developed. Specific kinds of analysis can be supported either by the dedicated language-specific functions (e.g., analysis of IT landscape heterogeneity), ADOxx's query mechanism, or by manual exploration and navigation through the created diagrams.

5 Conclusions

MEMO4ADO implements a selected and revised subset of the various MEMO DSMLs. It enables the user to develop an integrated, multi-perspective model of an enterprise, covering business processes, goal systems, IT infrastructures, and several other areas. To improve ease of use, and to overcome certain discrepancies between the architecture of MEMO and

ADOxx, several simplifications of the original design had to made in the implementation of MEMO4ADO. But the general character of MEMO has been retained in the tool.

Since the tool implements large portions of MEMO, it provides a platform to construct rich, detailed, and closely integrated multi-domain models of organizations and corresponding information systems. As illustrated previously, these models serve as a basis to answer a host of multidisciplinary analysis questions, especially pertaining to the integration of business and IT. Therefore, MEMO4ADO represents a versatile tool for IS professionals to adress the needs of a large variety of projects. So far as we know, its conceptual scope exceeds that of any other available enterprise modeling tool.

A further target user group of MEMO4ADO are students at the bachelor's and master's level. MEMO4ADO provides an accessible environment for students to familiarize themselves with the notion of multi-perspective enterprise modeling, and to explore the benefits of an integrated view of the organization. Another feature primarily directed to students is the integration of the 'basic' languages like ERM, DFD, and UML class diagrams. This capacity helps illustrate the advantages of using several modeling languages in tandem, and in the context of a more extensive enterprise model. We have been successfully using MEMO4ADO in different bachelor's and master's courses for several years now, with app. 200 students using the tool per year.

However, it is fair to say that precisely this conceptual complexity also remains the primary challenge to the applicability of MEMO4ADO. While it has been a general design goal of every MEMO language to reflect the subtleties and nuances of the subject, the cost has been a still rather long learning curve. To relax this conflict remains an objective of future work. Already existing features bearing on this issue include, for example, the ability to choose between different diagram detail levels.

MEMO4ADO remains under active development. We are currently working on a new version encompassing additional modeling languages, such as a new version of ITML, as well as new analysis functionalities.

Bibliography

[BF16] Bock, Alexander; Frank, Ulrich: Multi-perspective Enterprise Modeling—Conceptual Foundation and Implementation with ADOxx. In (Karagiannis, Dimitris; Mayr, Heinrich Christian; Mylopoulos, John P., eds): Domain-specific conceptual modeling, pp. 241–267. Springer, Cham, 2016.

[Bo15] Bock, Alexander Christopher: Beyond Narrow Decision Models: Toward Integrative Models of Organizational Decision Processes. In: 17th IEEE Conference on Business Informatics (CBI 2015). IEEE Computer Society, pp. 181–190, 2015.

[FB20] Frank, Ulrich; Bock, Alexander: Conjoint Analysis and Design of Business and IT: The Case for Multi-Perspective Enterprise Modeling. In (Kulkarni, Vinay; Reddy, Sreedhar; Clark, Tony; Barn, Balbir, eds): Advanced Digital Architectures for Model-Driven Adaptive Enterprises, pp. 15–45. IGI Global, 2020.

[FK13] Fill, Hans-Georg; Karagiannis, Dimitris: On the Conceptualisation of Modelling Methods Using the ADOxx Meta Modelling Platform. Enterprise Modelling and Information Systems Architectures, 8(1):4–25, 2013.

[FKHdK21] Frank, Ulrich; Kaczmarek-Heß, Monika; de Kinderen, Sybren: IT Infrastructure Modeling Language (ITML): A DSML for supporting IT management. Technical report, ICB-Research Report, 2021.

[Fr94] Frank, Ulrich: Multiperspektivische Unternehmensmodellierung: Theoretischer Hintergrund und Entwurf einer objektorientierten Entwicklungsumgebung. Oldenbourg, München, 1994.

[Fr11a] Frank, Ulrich: MEMO Organisation Modelling Language (1): Focus on Organisational Structure. ICB-Research Report 48, University of Duisburg-Essen, Essen, 2011.

[Fr11b] Frank, Ulrich: MEMO Organisation Modelling Language (2): Focus on Business Processes. ICB-Research Report 49, University of Duisburg-Essen, Essen, 2011.

[Fr11c] Frank, Ulrich: The MEMO Meta modeling Language (MML) and Language Architecture. 2nd Edition. ICB-Research Report 43, University of Duisburg-Essen, Essen, 2011.

[Fr14] Frank, Ulrich: Multi-Perspective Enterprise Modeling: Foundational Concepts, Prospects and Future Research Challenges. Software & Systems Modeling, 13(3):941–962, 2014.

[FS98] Ferstl, Otto K.; Sinz, Elmar J.: Modeling of business systems using the Semantic Object Model (SOM): A methodological framework. In (Bernus, Peter; Mertins, Kai; Schmidt, Günter, eds): Handbook on Architectures of Information Systems, International Handbooks on Information Systems, pp. 339–358. Springer, Berlin and Heidelberg, New York, 1998.

[GF10] Gulden, Jens; Frank, Ulrich: MEMOCenterNG – A full-featured modeling environment for organisation modeling and model-driven software development. In: Proceedings of the 2nd International Workshop on Future Trends of Model-Driven Development (FTMDD 2010). 2010.

[OFK15] Overbeek, Sietse; Frank, Ulrich; Köhling, Christian Alexander: A Language for Multi-Perspective Goal Modelling: Challenges, Requirements and Solutions. Computer Standards & Interfaces, 38:1–16, 2015.

[Sa14] Sandkuhl, Kurt; Stirna, Janis; Persson, Anne; Wißotzki, Matthias: Enterprise Modeling: Tackling business challenges with the 4EM method. The enterprise engineering series. Springer, Berlin [u.a.], 2014.

[St12] Strecker, Stefan; Frank, Ulrich; Heise, David; Kattenstroth, Heiko: MetricM: a modeling method in support of the reflective design and use of performance measurement systems. Inf. Syst. E Bus. Manag., 10(2):241–276, 2012.

[Za87] Zachman, John A.: A framework for information systems architecture. IBM Systems Journal, 26(3):276–292, 1987.