Graphical DSL Devoted to Ease the Application of Model-Driven Engineering in Petrochemical Industry Automation *

Jaqueline de Souza Moura[1], Carlos Eduardo Xavier Correia[2], Karina Casola[1], Fabio Paulo Basso[1], and Leandro Buss Becker[2]

2

[1]Federal University of Pampa, Alegrete, Brazil
https://unipampa.edu.br
[2]Federal University of Santa Catarina, Florianópolis, Brazil
https://ufsc.br/

Abstract. Software design for the petrochemical automation industry should follow the best available practices so that a high level of quality is achieved, together with modern practices that cope with reusability and automatic model transformation. In previous work, the M4PIA infrastructure was developed, which allows using Model-Driven Engineering (MDE) to support the pre-configuration of petrochemical industrial plants. Initially, the adopted models were represented only in an EMF hierarchical tree. However, an evaluative study showed that understanding such models is very hard from the perspective of the stakeholders, resulting in significant losses in the process. With this motivation, a graphical Domain Specific Language (DSL) was developed aiming to increase M4PIA's models readability. It was also developed a diagram modeling tool and a component library in the Eclipse platform. To evaluate the semantic transparency of the elements proposed in our DSL, a quasi-experiment with domain experts was conducted. Precision between 75%-100%, F-measure between 86%-100% and recall between 88%-100% were obtained for each element.

Keywords: Petrochemical Industry · MDE · DSL · Components Library · Graphical Representation · Eclipse

1 Introduction

An automation system for oil and gas (O&G) operations involves manipulating (measuring and controlling) thousands of plant variables. The process variables that are part of the control system are: the controlled variables, which have desired values to maintain in the process; manipulated variables, which can be adjusted so that the controlled ones stay as close as possible to your desired values; and the disturbances, variables of the process that affect the controlled ones, but cannot be manipulated and often cannot be measured. Due to the complexity of these petrochemical scenarios, simulation and control of automation

* Authors thank the funding agencies CAPES and CNPq. Authors at UFSC are also supported by the PRH-ANP/MCT nº 2.1 program.

has great relevance for industry and academia, contributing to the development of more efficient and safer technologies. An example of an automation system within the O&G domain is presented in [1]

For instance, the Brazilian state oil company Petrobras uses tools like MPA [2] and EMSO [3] to represent simulation processes and to design control techniques. These tools are complementary and share common data, but they are not integrated. In order to provide the integration of data from the tools, in previous works the tabular Domain Specific Language (DSL) named M4PIA (Model-Driven Engineering for Petrochemical Industry Automation) [4] was developed. It provided infrastructure for editing and transforming models dedicated to simulation and control. Currently, M4PIA models can be transformed to MPA and EMSO target platforms, also supporting reverse engineering [5].

The present work introduces a graphical DSL designed to improve the usability of M4PIA. The motivation for this development emerged from the preliminary evaluation of M4PIA's first representation, which used the EMF element tree. It was exposed that the semantic connection between MPA and EMSO was not obvious, making it difficult to transport information between such models [4]. Even representing both models with the same technology for concrete syntax, the perception of the connection of common elements between the two metamodels remained difficult. Therefore, the use of a graphical DSL was advocated to provide better semantic transparency. Additionally, the graphical DSL provides a visual representation of the domain problem that can be easily understood through the visualization of the overall picture of the problem represented by the language [6,7]. It also enables rapid user-level interaction through different prototypes and representing them at various levels of abstraction [8].

Even though the graphical DSL brings many facilities to the modeling process, as discussed in [9], repetitiveness is still a recurring problem in the industry, especially O&G one, given the need to model the same plant for each platform or program that will operate it. Designers in each application area need to have knowledge of the characteristics and behaviors of each process and component that make up the plant to perform specific modeling for the purpose of their platform. Thus, this work also presents the development of a library containing the main equipment and also the most common functions within O&G automation.

The reminder parts of this paper are organized as follows. Section 2 discusses relevant related works. In section 3 and section 4 we will address, respectively, the proposed DSL and tool and ours methodology evaluation with the obtained results. Finally, section 5 presents our conclusions and future works directions.

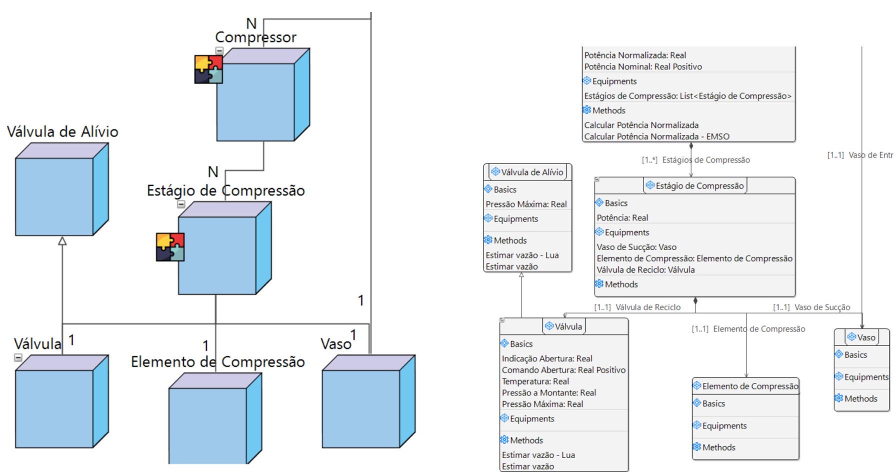
2 Related Works

In [10] it is addressed the increasing need for interoperability between systems and programs in the petrochemical industry. This branch is in constant development, so it is necessary that current programs are able to interact and operate in conjunction with legacies software. As main benefits resulting from the implementation of interoperability, the following are cited: easy and error-free data transfer, addition/removal of functionalities in applications facilitated, avoidance of commitment to programs exclusive to a single supplier company, and quality assurance. The paper provides five articles that look at interoperability in the downstream and upstream sectors of the industry. Finally, the authors expose that they believe in interoperability provided by frameworks for web services.

The work described in [11] idealizes an automated methodology for the development of multi-agent systems (MAS). With this objective, the i* modeling language was used to guide the development of MASs. However, such language has limitations regarding the capture of information necessary to design the architecture of MASs. Thus, to mitigate this problem, the UML language was applied, capable of supplying the deficiencies presented. Aiming at the cooperation of the models produced in both languages, transformations were used following the principles of model-driven engineering (MDE). In this way, the automation of the process makes it possible to make the mapping of the concepts used in the development of MASs more productive.

3 Proposed DSL and Developed Tool

During interviews with stakeholders, it was identified the need for multiple representations with different abstraction levels. To addressing such need, we developed two distinct graphical representations, the first is called Deployment Diagram (DD) and the other Conceptual Diagram (CD). Examples of these representations are shown, respectively, in Figures 1a and 1b.



(a) Deployment. **(b)** Conceptual.

Fig. 1: Example of language models.

The graphical editor was developed based in the elicited functional and non-functional requirements, as follows:

Functional Requirements: RF1. Manage the equipments, defining its attributes and functions; RF2. Compose equipment by other equipment; RF3.

Display composition and inheritance relationships; RF4. Display the multiplicity of relationships between equipment; RF5. Change equipment representation icons; RF6. Offer different viewpoints of the same model; RF7. Hide diagram components.

Non-Functional Requirements: RN1. Use the existing metamodel; RN2. Using the Eclipse infrastructure; RN3. Flexibility of use, allowing different ways of defining diagram items.

3.1 Modeling Tool

Hereafter, we have a description of the graphical representation of the modeling elements supported in our tool. In the DD, equipments are represented by a **node** called *CD Node Equipment*. By default, a 3D cube format was defined using a spatial notation. A child artifact was created being a border node *(Border CD Equipment Composition*), which has the shape of a *puzzle* to show the composition by other equipment(s). It is rendered only when the equipment is composed of other equipment(s). Furthermore, obtained from the semantic candidates, results of applying the filter by equipment type attributes, its display characteristic can be modified by the textual attribute *icon* of the candidates themselves.

In the CD, the equipment representation is done with a parent **container** (ED Container Equipment) that has another three child containers presented in a vertical stack. They are the compartments of basic attributes, equipment attributes, and methods. Each one has a child node represented in lists. In them, the respective properties are defined, such as the domain class and the expression of fundamental semantic candidates for the correct selection of these elements.

The relationships are represented by the **edges**. Both diagrams edges differ in the definition of the arrowhead decorator style. The DD does not use a decorator, while the CD uses a white triangular arrowhead for inheritance and a black diamond arrowhead for composition, as in the UML class diagram.

Filters and **layers** are used to automatically hide or add more information to the user. Relevant features have been defined for each diagram. Both diagrams have a filter to hide composition or inheritance relationships. For the conceptual diagram, there are filters that hide the compartments of an equipment, whether it be all compartments, the basic attributes compartment, the equipment attributes compartment, or the methods compartment.

3.2 Equipment and Function Library

The use of a library in the modeling stage is similar to reusing artifacts in the production of new software because both utilize tested and successful concrete objects for developing something new. In this sense, a library can provide several benefits not only in the modeling process but also in the generated model [12]. One of the benefits is the reduction of time required in the modeling stage due to the decrease in repetitive tasks and reduced need for testing, as the components used have little or no added errors. This allows for an increase in model output while maintaining adequate levels of quality and cohesion.

To find the equipment and functions, we used the MPA software libraries. Specifically, we utilized a total of ten MPA libraries provided by [13]. The attributes and methods that were feasible to implement were adapted for all components, whether they be equipment or functions.

During the tool's development process, the creation palette was split into five topics, each targeting a specific process within the modeling context. This division enhances the dynamic and cohesive use of the DSL in modeling processes, as each element category has a corresponding section for better user interaction. The division and its components or items can be analyzed in the Fig. 2.

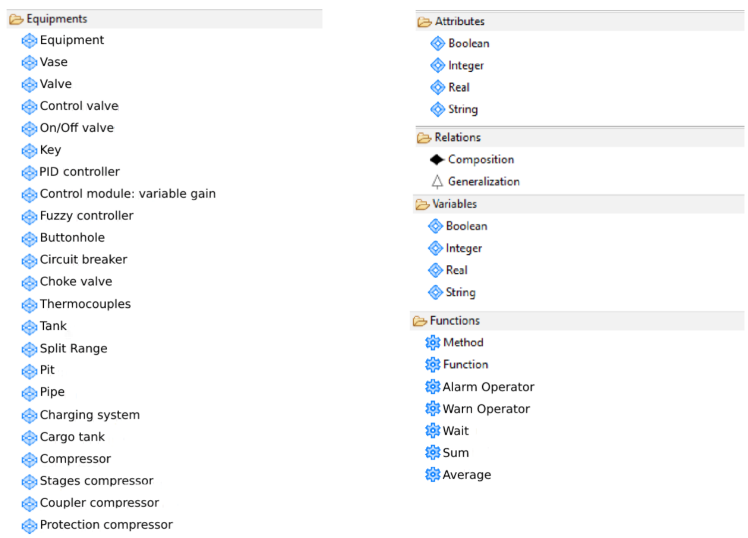


Fig. 2: Creation palette.

4 Evaluation

The evaluation of the proposed DSL used an experimental approach [14], consisting of five activities: planning, operation, data collection, analysis, and discussion of the results.

In the planning phase, the objectives, research questions, context, metrics, and participant selection were defined. Below, we can find the research questions that we sought to investigate:

- **RQ1.** What is the subject's perception of the usefulness and ease of understanding of the proposed DSL?
- **RQ2.** What is the subject's assessment of the representations of the elements specified in the proposed DSL?
- **RQ3.** Which model has greater acceptance: the conceptual diagram or the deployment diagram?

4.1 Planning and Execution

In the operational phase the evaluation design, instrumentation (profiling, elaboration of instrument 1 and 2, support materials, and consent and post-experiment terms) were defined.

Initially, the experiment was planned to occur in a face-to-face setting in a physical environment. However, due to the pandemic scenario of COVID-19, some limitations were imposed and led to adaptations in the planning. The alternative adopted was the use of a virtual room.

The selected participants are domain experts in petrochemicals who have worked or are working in simulation, control or operation of plants in the petrochemical industry. There were a total of nine initial participants, but one of them had to leave shortly after filling out the profile form and was excluded from the sample. Therefore, the sample is composed of the responses of eight subjects.

As a planning, before the execution of the tasks, the participants received: a contextualization material about gas compression system, the proposed DSL, and had five minutes to read it. Next, the participants received Instrument 1 with open-ended questions. Upon completion of this response, the link to the next Instrument 2 with the same questions in closed response format was received.

4.2 Results and Discussions

The syntax of the proposed language underwent a semantic evaluation with domain experts. The language elements obtained an accuracy between 75 and 100%, an F-measure between 86 and 100%, and a recall between 88 and 100%. This resulted in an average of over 90% for all metrics. Therefore, it was possible to verify the adherence and semantic understanding of the elements of each diagram. The method used to create the evaluation instrument was praised by the participants, mainly due to its objectiveness and fast execution, making it suitable for contemporaneity.

The representation adopted in the DD was positively evaluated by the participants. They reported that it was clean and highly understandable. The CD was praised for its detail and syntax that includes elements such as those from UML. Both solutions were pointed out for use in conjunction, i.e., complementary. To analyze the complete result of the experiment, see Table 1.

Conceptual Diagram: In instrument 1, the element with total transparency is Method, with 100% for all metrics. The others have a Precision of 88% and an F-measure of 93%, as there was one wrong answer. One participant indicated that the Equipment symbol was a basic attribute and vice versa. Another participant indicated that Composition and Inheritance determined where the arrow begins and ends, respectively. In instrument 2, the indices were raised for Equipment, Attribute, as well as Method, correctly identified by all, obtaining maximum metrics. One participant inverted the meanings of Composition and Inheritance, therefore scoring a Precision of 88% and an F-measure of 93%.

Deployment Diagram: In instrument 1, Inheritance demonstrated higher transparency, scoring 100% in all metrics. The least transparent is Multiplicity

1, which 75% precision and an F-measure of 86%. One participant interpreted its meaning as "at least one" and another as the order of execution. This same participant could not answer the meaning of N, which marks Multiplicity N, so their classification was P 100%, F 93%, and a recall of R 88%. Equipment and Composition obtained P 88% and F 93%. Equipment was identified as a class and Composition as an aggregation by one of the participants, while another indicated that Composite Equipment denoted inheritance. In instrument 2, this undergoes major changes. Equipment, Multiplicity 1, and N scored the highest metrics. One participant inverted the definitions of Composite Equipment and Composition, although they had correctly answered in the previous instrument, indicating possible lack of attention. Another participant made a mistake between the meanings of Composition and Inheritance. Consequently, Composition had the lowest classification, P 75% and F 86%, followed by Composite Equipment and Inheritance with P 88% and F 93%.

Table 1: Obtained results about the Conceptual and Deployment Diagrams.

Diagram	**Element**	**Instrument 1**			**Instrument 2**		
		P	**R**	**F**	**P**	**R**	**F**
Conceptual	Equipment	88%	100%	93%	100%	100%	100%
	Attribute	88%	100%	93%	100%	100%	100%
	Method	100%	100%	100%	100%	100%	100%
	Composition	88%	100%	93%	88%	100%	93%
	Inheritance	88%	100%	93%	88%	100%	93%
	Average	90%	100%	94%	95%	100%	97%
Deployment	Equipament	88%	100%	93%	100%	100%	100%
	Composite Equipament	88%	100%	93%	88%	100%	93%
	Inheritance	100%	100%	100%	88%	100%	93%
	Composition	88%	100%	93%	75%	100%	86%
	Multiplicity 1	75%	100%	86%	100%	100%	100%
	Multiplicity N	100%	88%	93%	100%	100%	100%
	Average	90%	98%	93%	92%	100%	95%

5 Conclusions and Future Works

This work aimed to enhance the usability of the previously developed M4PIA tool set, which intended for the using MDE within the O&G industry, providing bidirectional models transformations. In order to ascertain the proposals of this work, we conducted an experimental evaluation with the collaboration of the stakeholders themselves.

Based on the obtained results from the semantic evaluation, it can be stated that the proposed DSL has elements that are well understood by domain experts. However, there is still space for further improvement in defining some elements, especially regarding the understanding of certain concepts such as Composition and Inheritance, for example. Moreover, it is important to note that the evaluation was conducted with a limited number of participants, which may limit the generalization of the results. Therefore, it is recommended that future studies include a larger number of experts to further validate the proposed DSL.

As a future work, it is proposed to conduct a survey with the stakeholders to find the equipment and functions with more impacts and uses in the petrochemical industry in order to add content to the proposed library. It is also idealized to validate this library through tests and experiments with the stakeholders, thus seeking to obtain additions and ideas to improve the tool.

References

1. R. S. Gesser, R. Sartori, T. P. Damo, C. M. Vettorazzo, L. B. Becker, D. M. Lima, M. L. de Lima, L. D. Ribeiro, M. C. Campos, and J. E. Normey-Rico, "Advanced control applied to a gas compression system of an offshore platform: From modeling to related system infrastructure," *Journal of Petroleum Science and Engineering*, vol. 208, p. 109428, 2022.
2. E. Satuf, S. F. Pinto, and B. Q. Dias, "Automatic alignment system for pra-1 pumping platform (in portuguese)," in *5th Rio Automation Congress*, 2009.
3. R. d. P. Soares and A. Secchi, "EMSO: A new environment for modelling, simulation and optimisation," in *Computer Aided Chemical Engineering*, vol. 14, pp. 947–952, Elsevier, 2003.
4. T. P. Damo, L. B. Becker, and F. P. Basso, "Model-Driven Engineering Infrastructure and Tool Support for Petrochemical Industry Automation," *Advances in Science, Technology and Engineering Systems Journal*, vol. 4, no. 4, pp. 174–187, 2019.
5. M. V. Silva Cruz, T. P. Damo, and L. B. Becker, "Round-trip engineering for petrochemical industry automation," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 11824–11829, 2020. 21st IFAC World Congress.
6. E. Evans, *Domain-Driven Design: Tackling Complexity in the Heart of Software.* Addison-Wesley Professional, 2004.
7. S. Cook, G. Jones, S. Kent, and A. C. Wills, *Domain-Specific Development with Visual Studio DSL Tools.* Pearson Education, 2007.
8. P. De Smedt, "Comparing three graphical DSL editors: AToM3, metaedit+ and poseidon for DSLs," *Preprint, Submitted to Elsevier, University of Antwerp*, 2011.
9. M. S. Niklas Mellegård, "A Domain Specific Modelling Language for Specifying and Visualizing Requirements," *CAiSE2009*, 2009.
10. B. Braunschweig, "Software interoperability for petroleum applications," vol. 60, no. 4, pp. 587–596, 2005.
11. J. A. A. M. Carla Silva, Pedro Dias, "De arquitecturas organizacionais em i* a arquitecturas baseadas em agentes: Uma abordagem orientada a modelos," in *Anais do WER11 - Workshop em Engenharia de Requisitos*, (Rio de Janeiro, Brasil), pp. 357–368, Abril 2011.
12. J. L. Barros-Justo, F. Pinciroli, S. Matalonga, and N. Martínez-Araujo, "What software reuse benefits have been transferred to the industry? A systematic mapping study," *Information and Software Technology*, vol. 103, pp. 1–21, 2018.
13. T. I. (PUC-Rio), *MPA Programming Libraries from the Tecgraf Institute (PUC–Rio)*, 2018. `https://git.tecgraf.puc-rio.br/mpa/libs/pucrio-tecgraf`.
14. C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering.* Springer Science & Business Media, 2012.