Towards AI Assisted Domain Modeling

Christophe Feltus[1], Qin Ma[2], Hend*erik* A. Proper[1,2], and Pierre Kelsen[2]

[1] Luxembourg Institute of Science and Technology
[2] University of Luxembourg
christophe.feltus@list.lu, qin.ma@uni.lu,
e.proper@acm.org, pierre.kelsen@uni.lu

Abstract. A domain model provides an explicit knowledge representation of (selected aspects of) some domain of interest. The transition to the digital age results in an increased need for domain models that are machine understandable. We posit that, at the same time, there is an increasing need for non-experts (in modeling) to be able to create such models, or at least be able to understand the created models, and take *ownership* of their meaning and implications. This situation causes a 'modeling bottleneck' in that it is not reasonable to expect all non-experts to become modeling experts. This is where we turn to AI as an enabling technology to support non-experts in domain modeling related tasks; i.e. AI Assisted Domain Modeling. We foresee a symbiotic collaboration between human intelligence, symbolic AI and subsymbolic AI; essentially resulting in a triple-helix of human, symbolic, and subsymbolic intelligence.

The aim of this workshop paper is to structurally explore the potential role of (symbolic and subsymbolic) AI to support domain conceptualization. To do so, we will combine three perspectives on domain modeling: (1) a framework relating the different conceptions (harbored in the mind of a modeler) regarding the domain to be modeled, and the model itself, (2) the role of normative frames towards modeling activities, and (3) modeling as a structured dialogue between an (automated) system analyst and a domain expert.

1 Introduction

A domain model provides an explicit knowledge representation of (selected aspects of) some domain of interest. Such a domain of interest may e.g. involve an existing part of an enterprise, or an envisioned future situation, etc. [38, 39]. Software engineering, information systems engineering, and enterprise engineering, have a rich tradition in the use of different kind of domain models. This includes a.o.: enterprise (architecture) models, business process models, organizational models, information models, software models, ontologies, knowledge graphs, etc. In line with [38, 39], we consider each of these kinds of models as valued members of the larger family of *domain models*.[3]

[3] Not all domain models are *conceptual models*. As discussed in e.g. [38], a distinction could be made between *conceptual* domain models and *computational design* models,

In our present day society, we can observe a strong increase in the role/use of knowledge-intensive computing technology, including *(explainable) AI*, *data science*, and *digital twins* [29]. Meanwhile, such knowledge-intensive computing technologies have permeated virtually all facets of society. From manufacturing, logistics, finance, health, to space exploration. With this increase, also comes an increase in the need to capture relevant domain knowledge by means of domain models in a format which is understandable by both humans and 'machines'.[4]

As a result, there is an increasing need for non-experts (in modeling) to be able to create such models, or at least be able to understand the created models, and take *ownership* of their meaning and implications. The authors of [43] also observe how modeling increasingly becomes embedded in everyday work. The latter makes it inevitable for non-experts (in modeling) to also be able to engage in modeling activities. This situation causes a 'modeling bottleneck' in the sense that it is not reasonable to expect all non-experts to become modeling experts.

Moreover, supporting modeling processes is one of the major challenges for domain modeling, as observed in [36]. This challenge has fueled earlier efforts to make modeling strategies more explicit [24, 25, 26], as well as experiments with the concept of *natural modeling* [8, 47], which has also been echoed by the more recent notion of *grassroots modeling* [43].

The authors of [43] suggest the use of *assistive technologies* to support modeling activities by non-experts. Inspired by this, we prefer to speak about *Assisted Domain Modeling*. Furthermore, we turn to AI as an enabling technology to drive the needed assistance, hence AI Assisted Domain Modeling. More specifically, we foresee a symbiotic collaboration between human intelligence, symbolic AI and subsymbolic AI for assisting domain modeling, which essentially results in a triple-helix of human, symbolic-driven and subsymbolic-driven intelligence.

A critical aspect in the creation and interpretation of domain models is the conceptualization of the domain that is (to be) captured in the model [39]. Therefore, we initially focus on this aspect.

In line with this, the aim of this *workshop paper*, is to work towards a structural exploration of the potential role of AI to support domain conceptualization. To do so, we will combine three existing perspectives regarding domain modeling: (**p1**) a framework [39, 38] relating the different conceptions (as harbored in the mind of an modeler) regarding the domain to be modeled, and the model itself, (**p2**) the role of normative frames [36] towards modeling activities, and (**p3**) viewing modeling as a structured dialogue between an (automated) system analyst and a domain expert [9]. The integration of these perspectives will enable us to more closely investigate the potential role of AI to support domain conceptualization. In this paper, we will also take first steps towards the latter.

An important disclaimer we need to make here, is that for now, we do not (yet) consider a situation where multiple modelers collaborate in the creation of a domain, i.e., collaborative modeling.

where the latter involve 'compromises' needed to support computational (design) considerations to e.g. support simulation, animation, or even execution of the model.

[4] But this does not necessarily imply that these models should be executable.

The remainder of this paper is structured as follows. In Section 2 we present an integrated view on domain modeling, which combines perspectives **p1** and **p2**. In moving towards AI Assisted Domain Modeling, Section 3 then complements this with perspective **p3** to arrive at a framework to understand/position the key activities involved in domain modeling in general, and conceptualization in particular. Based on this, Section 4 provides a short reflection on the possible role of AI to support domain modeling. Section 5 and 6 then explore the potential role of symbolical and subsymbolic AI, respectively, to support/drive the involved activities. In Section 7 we conclude the paper, while also reflecting on next steps towards the elaboration of the presented framework, as well as concrete experiments towards AI support.

2 Understanding Domain Modeling

In line with [7, 38, 39], we consider a *domain model* to be: A *social artifact* that is *acknowledged* by an *observer* to *represent* an *abstraction* of some *domain* for a particular *purpose*.

A model is a *social artifact* in the sense that its role as a model should be recognizable by (a) collective agent(s) [39]. For this reason, it should exist outside of our minds. In our field of application, this artifact often takes the *form* of some 'boxes-and-lines' diagram. More generally, however, domain models can, depending on the *purpose* at hand, take other forms as well, including text, mathematical specifications, games, animations, simulations, and physical objects. It is ultimately the observer who needs to *acknowledge* the fact that an *artifact* is indeed a model of the domain, for the given *purpose*. Since a model is the representation of an *abstraction* of the domain, some 'details' of the domain are consciously left out, in line with the *purpose* of the model.

In the context of modeling, and following [39], we suggest to make a distinction between two kinds of (composed) thoughts that the observer may have about the domain: *conceptions* and *perceptions*. When actors observe a domain, they will obtain (through their senses) a *perception*. They may than be able to interpret, structure, and/or further abstract this *perception* to form a *conception* in terms of concepts and relations among the concepts.

In *perceiving*, the multitude of facets and nuances of the world around us, forces us to apply filters. When creating a conception from our perception, we tend to filter even further, consciously leaving out details in order to be able to focus on what we think is important (in particular when creating models). This is also where we apply our 'hard-wired' ability to classify our observations and make generalizations [31, 32].

As a consequence of the above, a modeler (be it an expert modeler or a non-expert modeler) needs to harbor (at least) four conceptions [39] in their mind: (1) a 'full' conception of the domain (as they 'sense' it); (2) a conception of the purpose for the model; (3) a filtered focused conception of the domain, based on the purpose of the model; and (4) a conception of the artifact that is (to be) the model representing the focused domain conception. These four conceptions are shown (as circles) within the darker gray area, labeled Foreground Conceptions, of Fig. 1. The rectangles represent the externally (to the actor) observable Purpose, Domain, etc. Fig. 1 also shows how the (conception of the) *purpose* of the domain model, influences/modifies the observation and focus. This structure is based on [39], where we have now modified some of the terms to better clarify the fact that the *domain focus* is the conception that results after the *purposeful* filtering/abstraction of the original 'raw' conception of the domain.

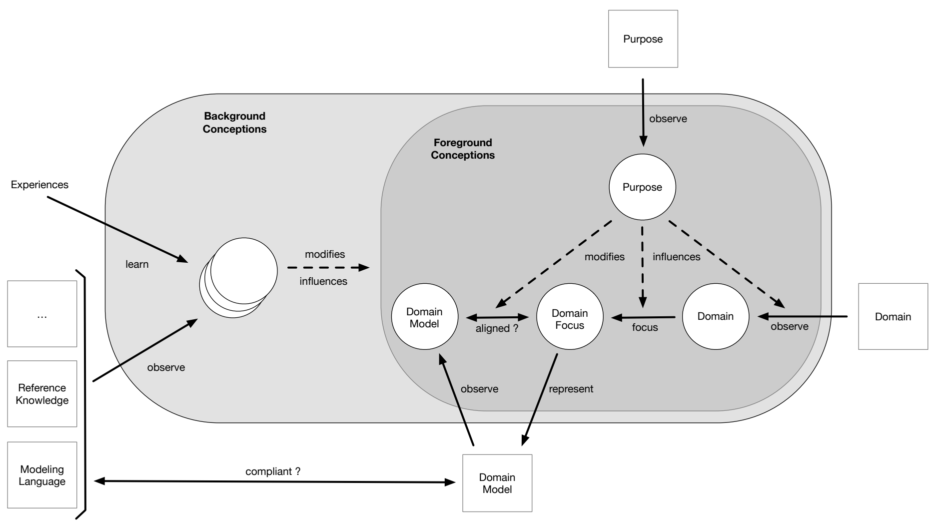

Fig. 1. Conceptions playing a role in domain modeling

One of the challenges when expressing the *domain focus* in terms of a *domain model* is to make sure that the latter is 'shaped' in such a way that it corresponds to the former. However, since the *domain model* is external to the modeler, this operationally means that the conception of the *domain model* needs to be aligned to the *domain focus* conception.

It should be noted that Fig. 1 actually captures two possible scenarios: (1) a modeler creating a model of a domain, in line with the purpose, and (2) a (non-expert) modeler who is asked to acknowledge a given (i.e. created by another modeler) model as being a valid model (of the domain, in line with the purpose).

The above discussion covers perspective **p1**, concerning different conceptions involved in domain modeling. This takes us to **p2**, i.e., the role of normative frames. As argued in [36, 39], modeling is influenced/modified by several *normative frames*, including the used design/engineering framework, the (to be used) modeling language, personal biases, etc. These normative frames influence/modify the Foreground Conceptions as shown in the dark gray area of Fig. 1. When the modeler has an explicit understanding of the normative frames they (consciously or unconsciously) use, this results in additional conceptions. In Fig. 1, this is shown as the Background Conceptions pertaining to e.g., the definition of a modeling language, reference models, gained experience, etc.

3 Supporting Domain Modeling

In this Section, we look at the question of how a modeling processes can be supported, more specifically in terms of the concept of a *modeling assistant*. In doing so, we will not yet make a choice if such a *modeling assistant* should involve human intelligence or any form of AI. This question will be considered in more detail in the next Sections.

In [9, 24, 25, 26], the authors take the perspective (**p3**) that a modeling process involves a structured dialogue between a *domain expert* and a *system analyst*. We contend that such a *structured dialogue*, which can also be seen as a kind of a 'think aloud protocol', provides a good vantage point to support domain modeling, in particular, when it is a (machine) observable dialogue. The ambitions, as reported in [9, 24, 25, 26], also point towards strategies behind such a structured dialogue. Moreover, the perspective on a modeling process as a structured dialogue, also resulted in hypotheses on the competences required from the participants in a modeling process [15].

We will, therefore, take the *structured dialogue* perspective as a starting point in identifying ways to support modeling processes. However, since we are concerned with the challenge of providing modelers in general (be they experts in modeling or not) with more support for their modeling activities, we actually prefer not to speak about a distinction between a *domain expert* and a *system analyst*, but rather speak about a *modeler* and a *modeling assistant*.

It should also be noted that the assumptions as made in [15] regarding the roles and competencies required for a modeling process, 'silently' puts the onus of the initial *domain conceptualization* on the domain expert. For instance, it is assumed that "*domain experts can provide any number of significant sample sentences in relation to relevant information objects*". This implies, however, that the domain expert must already have a clear *domain conception* [39] in their mind. As such, we certainly acknowledge the work done in [15], while at the same time being more ambitious: *Can domain experts be supported in conceptualizing their domain of expertise?* In particular, when moving from the initial *domain conception* to the conception of a *focused domain* that is in line with the modeling *purpose* at hand.

Building on the concept of *structured (modeling) dialogues*, we propose the framework as shown in Fig. 2. At the top, we find, following Fig. 1, a modeler observes a domain, and then given a purpose, aims to produce (or validate) a domain model. At the bottom, we find a modeling assistant. For a modeling assistant to aid a modeler, 'it' should have an understanding of the foreground and background conceptions as held by the supported modeler (resulting in the P', D', DF' and DM' 'shadow' conceptions). We contend that the only way for the modeling assistant to develop this understanding, is to (1) observe the actions of the modeler and develop a Modeler profile; (2) conduct a structured dialogue regarding the different conceptions held by the modeler (see the rectangle labeled with Supporting dialogues); and/or (3) co-create the actual Domain model. Ideally, a modeler would (be nudged to) use a 'think-aloud' strategy when modeling a domain. More specifically, making their considerations explicit (preferably in

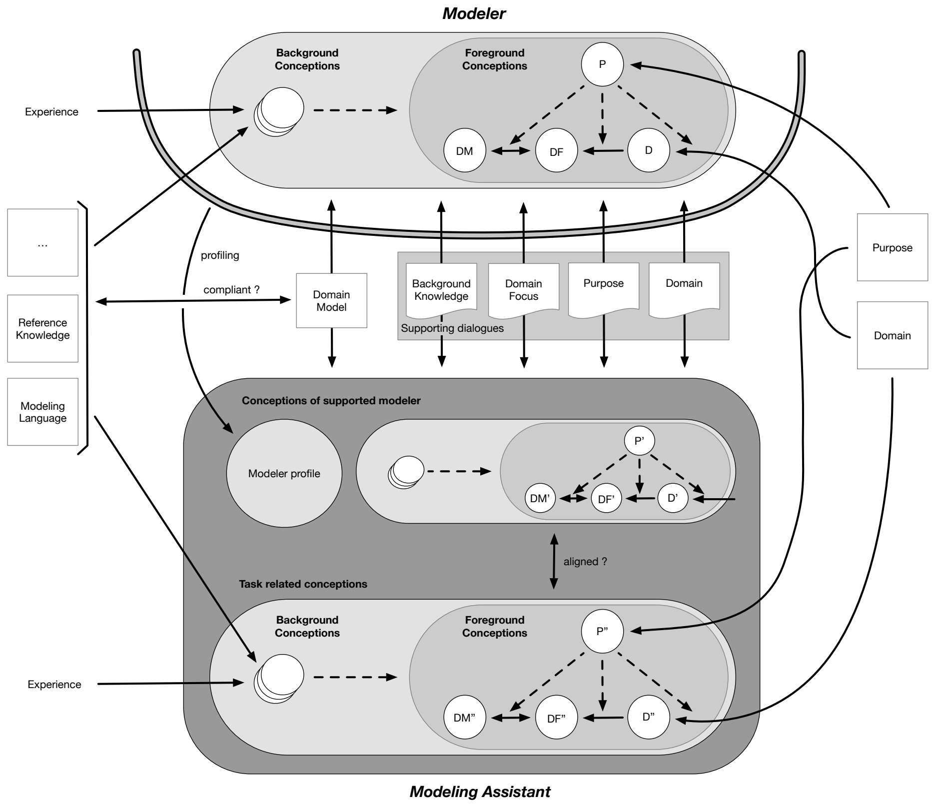

Fig. 2. Support for domain modeling

a machine understandable format), will be beneficial in driving the structured dialogue(s) needed for the modeling assistant to derive an approximation of the conceptions held by the modeler.

Since the modeling assistant is likely able to observe/read itself the modeling purpose, the domain to be modeled, the definition of the modeling language to be used (potentially even in a machine readable format), background knowledge, etc., the modeling assistant may also harbor a conception of the purpose, the domain, the domain focus, etc., in line with the modeling task at hand, resulting in the conceptions P'', D'', DF'' and DM''.

The combination of these, allows the modeling assistant to aid the modeler in their task to create a domain model and/or understand/validate/evaluate an existing domain model in relation to its purpose and the domain.

4 Towards AI support

When using AI as an enabling technology to realize modeling assistants, the conceptions as (to be) harbored in 'the mind' of the AI assistants will take the form of trained statistical networks and/or more traditional machine understandable representations. In [12], the author provides a (not exhaustive) categorization of modeling related 'problems' that may be supported by means of AI. Using these categories as a base, combined with Fig. 2, this would give rise to the following first identification of potential AI support.

Gathering background knowledge:

- **Pattern mining in models**: Reusable similar modeling components are identified by mining typical modeling patterns [19].
- **Finding matches between modeling constructs**: Identification of similar modeling constructs in different models [3].

Supporting structured modeling dialogues:

- **Modeling guidance**: Syntactic, semantic, or pragmatic guidance is provided during the step-by-step refinement/specialization of the models by the modeler [40, 37, 35, 42, 1, 2, 10, 6].
- **Explicit strategies for modeling** [26, 23].

Correction and validation of models:

- **Automatic modeling and model correction**: Models are automatically corrected, or developed by automated planning [21].

Model representation and interpretation:

- **Model-to-Text, Text-to-Model or Picture-to-Model**: Natural texts, dialogues, and pictures are transformed into models and vice versa [23, 17, 46, 22, 41].

In the next two Sections, we discuss the potential AI contributions related to these categories in more detail.

5 A Potential Role for Symbolic AI

Symbolic AI methods represent knowledge using a symbol-based representation [13]. For instance, in the form of graphs, logic formulas, symbolic rules, and relies on formal operations, such as inference and production, over the symbolic representation of knowledge to simulate mental or cognitive operations.

In [20], John Haugeland coined the term GOFAI ("Good Old-Fashioned Artificial Intelligence"). One may argue that (good old) symbolic AI is 'old-fashioned', especially when compared to the present advances in 'data- and statistics-driven' subsymbolic AI techniques. However, there is still considerable value in symbolic AI. For example, it is good for addressing problems that need logical inference on knowledge representation. Symbolic AI techniques, such as problem solving in terms of constraints satisfaction, Natural Language processing (NLP), and logical inference, are still among the core functionalities of current AI systems. Moreover, symbolic AI can provide good solution in circumstances where statistical approaches fall short, e.g., when there is not enough data available. Symbolic reasoning also provides inherently more transparency regarding the reasoning process, thus enabling explainable AI more easily.

In fact, many researchers start to resort to the combination of both kinds of AI when designing a solution for their problems. For example, [42] exploits both rule-based NLP techniques and data-driven ML techniques to design a web-based bot that can automatically extract domain models (class diagrams) from domain problem descriptions, and demonstrates that the combined solution out-performs others where only one style of AI techniques is applied.

Let us come back to the process of assisted domain conceptualization as depicted Fig. 2 and point out some example steps for which symbolic AI can provide help. The process starts with both modeler and modeling assistant observing the domain to be modeled, the purpose of modeling, any contextual normative frames, and forming a conception of each. This step involves the three generic mental operations (acts) coined by St. Thomas Acquinas [16]: defining concepts, pronouncing judgments (relations) between concepts, and reasoning new knowledge from knowledge existing in a knowledge base constructed by the first two acts. We acknowledge that the domain, the purpose, and the normative frames can exist in various formats, e.g., as a physical object, a document in natural language, a vocal agreement, or a picture. Depending on the format and the sensing channels used by the modeler and the modeling assistant during the observation, various rule-based AI techniques such as NLP (Natural Language Processing), and pattern recognition, can be exploited.

When both modeler and assistant form the individual conceptions, they communicate (through dialogues or Q&A sessions) with each other to understand counterpart's conceptions, to validate, to update, to improve, and eventually to agree upon on an aligned conceptualization of the referents, uttered in the form of artifacts. In this aspect, if the assistant is a computer, conversational AI techniques, such as dialogue systems, conversational agents, and chat-bots, will be the areas to resort to. Such systems can e.g. be used to operationalize the rule-based modeling strategies as suggested in e.g. [9, 24, 27, 18, 30].

Moreover, existing symbolic AI approaches, such as recommender systems can be leveraged to provide guidance for structured modeling dialogues. As an example, the work of Agt-Rickauer et al. [1, 2] demonstrates how symbolic AI techniques can be jointly exploited to support domain modeling. More specifically, they use rule-based NLP techniques and knowledge representation techniques to access domain knowledge from heterogeneous sources, including textual data sets and existing knowledge bases (such as ontologies), and make use of a recommender system to offer semantic (vs. syntactical) modeling support/ recommendation to modelers in various modeling scenarios, by capitalizing on the domain knowledge gathered in the first part.

6 A Potential Role for Subsymbolic AI

In this Section, we review some *subsymbolic* AI approaches that can potentially be used in the context of a modeling assistant.

Principal Component Analysis (PCA) is a well-known ML technique [28] that aims to reduce the number of dimensions of large data without losing too much information. This age-old technique is now embedded in ML libraries[5]

[5] `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html`

and now benefits of more computational power than ever. It is based on the fact that many types of vector spatial data are compressible, and compression can be performed more efficiently by sampling. The main advantages of PCA are better data visualization and optimization of resources when the PCA is used to prepare the data before processing by another algorithm.

There is little scientific contribution which exploits PCA for modeling purposes. Nevertheless, [10] illustrates the use of PCA for *modeling assistance* (towards *model representation & interpretation*) by investigating the effect of structural complexity on the understandability of state-chart diagrams. The authors identified three types of complexity, namely: (1) the size and control flow complexity of the state-chart, (2) the actions performed when entering or leaving a state, and (3) the sequence of actions performed within the state. Based on this, elaborated a prediction model to understand the impact of the complexities on the state-chart understandably. To identify the components of structural complexity, [10] uses the Principal Component Analysis statistical data reduction method.

Clustering algorithms[6] aim to gather types of machine learning unsupervised algorithms that group various (unlabelled) objects into clusters based on the object's features.

Consider, for instance, a database of activities (unlabelled objects) described by a set of features like the *activities' inputs and outputs, the stakeholders involved in the activity, the activity duration, or the previous and next activities*, etc. This database may be represented as an input matrix for a clustering algorithm and the output vector of that algorithm is a set of n clusters of activities like, e.g., *send a message, validate a reply, make an order*, etc. These clusters may consist in a relevant information to support the conceptualization of generic activities to be used afterwards to elaborate, for example, a process model.

Some famous clustering algorithms are K-Means, Mean-Shift, Agglomerative Clustering, DBScan, etc. In that list, K-Means [33] aims to split the observed data into K clusters in which each observation is included in the cluster with the nearest mean. K-Means consists by the way to minimize the within-cluster sum-of-squares value.

Clustering naturally contributes to the modeling tasks of *pattern mining in models* and *finding matches between modeling constructs.* Additionally, clustering may also contribute to other tasks such as the *automatic modeling and model correction.* For instance, [6] exploits clustering algorithms for *automatic refactoring* of UML diagrams. Therefore, the authors address the software refactoring at the design time and at the UML diagram level. At the software-side, refactoring means *changing the internal structure of a software to make it easier to understand and cheaper to modify without changing its observable behavior* [14]. In [19], clustering is used to support *pattern mining in models.* The authors exploit the RPST (Refined Process Structure Tree) to derive trees of nested process fragments from different models. Patterns are afterward determined among these nested fragments.

[6] `https://scikit-learn.org/stable/modules/clustering.html`

Classification algorithms[7] involves supervised algorithms that, when trained, can classify objects based on their features. A classical example is the classification of the Iris flower. A machine learning model is trained with a dataset containing three types of labeled input corresponding to Iris species (Iris setosa, Iris virginica and Iris versicolor) and described by four features: length and width of sepals and petals. When trained, the model allows identifying new samples based on its features only.

Transposed to the modeling task of e.g. *finding matches between modeling constructs*, a machine learning model can be trained to, for instance, improve the classification of business activities into three types of ArchiMate [5] elements (*business process*, *business function* and *business task*) based on features such as *the duration of the activity*, *the level of description of this activity*, *the number of stakeholders involved*, etc. When a new activity must be conceptualized, the trained machine learning model can then help in classifying it in *business process, business function* or *business task*.

Convolutional neural networks (CNNs) [4] are used for image classification (e.g., detect objects in picture). The input of a CNN is very often an image (but it may be any type of data object) represented by an array of pixels, and the output is the class to which the image has a probability of belonging. Such algorithms are made up of two parts. The first part consists of a feature extractor and the second part consists of a set of fully connected layers which takes the feature vectors as input and maps them to a final layer where each neuron corresponds to the predicted class (e.g., a dog, a cat, a circle, a square, etc.).

CNNs can play a role with regards to *Picture-to-Model* tasks. For instance, in the recent paper [17], the authors propose a CNN based tool to classify an input images as UML class diagrams and non-UML class diagrams or in [46], the authors propose a application to recognize the structure of business process models drawn on paper and whiteboard. Thereby, AI is exploited to create digital versions of the models. Yet another interesting contributions is the classification of class diagrams that has been investigated by means of image classification in [22]. After applying image classification to class diagrams, the features extracted were tested with five machine learners to assess the accuracy of their classification.

Recurrent neural networks (RNNs) [44] are often used for speech recognition. At times, the domain to be modeled may also have been described (partially) by speech. E.g., a process owner describing a process during an interview. This oral description needs first to be transcribed into a text.

In those cases, the analysis output of the first piece of data (e.g. a word) needs to be considered for the analysis of the subsequent ones. The output of an RNN can be analyzed further by other algorithms in order to extract the important elements that it contains such as, for the description of a process: *the tasks, the stakeholders, the inputs* or *outputs*, etc. To do this, other techniques may be used such as the word embedding which is a *language modeling tool*

[7] `https://scikit-learn.org/stable/supervised_learning.html`

that consists in learning the representation of words from a text based on their semantics, on the context, and/or on their occurrences in a text.

Word embedding, together with the transformer mechanism, is already used for sentiment analysis, question-answering, and text-summarization tasks. It could also be extended for domain conceptualization.

In the field of text-to-model mining, on overview of the current state of the art is provided in [41]. The paper presents different approaches focusing on business process models. These approaches are analyzed and compared against each other both at a theoretical and a technical point of view.

Reinforcement Learning (RL) involves software agents learning to react on their own to an environment that they do not yet know [45]. In order to learn how to react, agents make decisions and take actions with the objective of accumulating rewards while avoiding errors.

Despite RLs' exponential deployment [11], as far as our knowledge, there is not yet a sound scientific contribution that addresses the conceptualization with reinforcement learning, we believe that the discovery of new concept considering this AI technique makes perfect sense. Indeed, the activity of conceptualization is a gradual activity, during which the modeler learns to discover new concepts by improving his reasoning from his successes and his mistakes.

This learning method is therefore naturally assimilated to the learning algorithms offered by reinforcement learning which also consists of discovering information with back and forth steps, and trial and error. It would therefore not be surprising that further research develops within a few years in this area and contributes to the activities of *pattern mining in models* and/or *finding matches between modeling constructs.*

7 Conclusion

The aim of this workshop paper, was to work towards a structural exploration of the potential role of (symbolic and subsymbolic) AI to support domain conceptualization. In doing so, we first combined three existing perspectives on domain modeling: (1) a framework relating the different conceptions (harbored in the mind of a modeler) regarding the domain to be modeled, and the model itself, (2) the role of normative frames towards modeling activities, and (3) modeling as a structured dialogue between an (automated) system analyst and a domain expert. This resulted in a general understanding of the (conceptualization) activities involved in domain modeling (Fig. 1), as well as a first understanding of the core functionalities that would be needed from a *modeling assistant* (Fig. 2). In terms of this, we then provided an initial survey of the potential role of symbolic and subsymbolic AI in supporting domain modeling, where we foresee a symbiotic collaboration between human intelligence, symbolic AI and subsymbolic AI.

As the next step, we aim to (1) further elaborate our understanding of the potential role of modeling assistance (i.e. elaborate on Fig. 2, while also including a collaborative modeling perspective), (2) elaborate the survey of available AI approaches and techniques that may be used to provide modeling assistance, and (3) initiate a series of experiments with different strategies/techniques for modeling support.

References

1. Agt-Rickauer, H.: Supporting domain modeling with automated knowledge acquisition and modeling recommendations. Ph.D. thesis, Technical University of Berlin, Germany (2020)
2. Agt-Rickauer, H., Kutsche, R.D., Sack, H.: Domore - A recommender system for domain modeling. In: Proceedings of the 6th International Conference on Model-Driven Engineering and Software Development, MODELSWARD 2018, Funchal, Madeira - Portugal, January 22-24, 2018. pp. 71–82. SciTePress (2018)
3. Ahmed, J., Huang, M.: Classification of role stereotypes for classes in uml class diagrams using machine learning (2020), https://gupea.ub.gu.se/handle/2077/67955
4. Albawi, S., Mohammed, T.A., Al-Zawi, S.: Understanding of a convolutional neural network. In: 2017 Int. Conf. on Engineering and Technology (ICET). pp. 1–6 (2017)
5. Band, I., Ellefsen, T., Estrem, B., Iacob, M.E., Jonkers, H., Lankhorst, M.M., Nilsen, D., Proper, H.A., Quartel, D.A.C., Thorn, S.: ArchiMate 3.0 Specification. The Open Group (2016)
6. Baqais, A., Alshayeb, M.: Automatic refactoring of single and multiple-view uml models using artificial intelligence algorithms. In: Doctoral Consortium (MODELSWARD 2016). pp. 3–8. INSTICC, SciTePress (2016)
7. Bjeković, M., Proper, H.A., Sottet, J.S.: Embracing pragmatics. In: Yu, E.S.K., Dobbie, G., Jarke, M., Purao, S. (eds.) Conceptual Modeling – 33rd International Conference, ER 2014, Atlanta, GA, USA, October 27-29, 2014. Proceedings. LNCS, vol. 8824, pp. 431–444. Springer (2014)
8. Bjeković, M., Sottet, J.S., Favre, J.M., Proper, H.A.: A framework for natural enterprise modelling. In: IEEE 15th Conference on Business Informatics, CBI 2013, Vienna, Austria, July 15-18, 2013. pp. 79–84. IEEE Computer Society Press (2013)
9. Bommel, P.v., Hoppenbrouwers, S.J.B.A., Proper, H.A., Weide, T.P.v.d.: Exploring modelling strategies in a meta-modelling context. In: Meersman et al. [34], pp. 1128–1137
10. Cruz-Lemus, J.A., et al.: The impact of structural complexity on the understandability of uml statechart diagrams. Information Sciences **180**(11), 2209–2220 (2010)
11. Feltus, C.: AI's contribution to ubiquitous systems and pervasive networks security-Reinforcement Learning vs. Recurrent Networks. J. Ubiquitous Syst. Pervasive Networks **15**(02), 1–9 (2021)
12. Fettke, P.: Conceptual modelling and artificial intelligence: Overview and research challenges from the perspective of predictive business process management. In: Companion Proceedings of Modellierung 2020 Short, Workshop and Tools & Demo Papers co-located with Modellierung 2020, Vienna, Austria, February 19-21, 2020. CEUR Workshop Proceedings, vol. 2542, pp. 157–164 (2020)
13. Flasiński, M.: Introduction to Artificial Intelligence. Springer (2016)
14. Fowler, M.: Refactoring: improving the design of existing code. Addison-Wesley Professional (2018)
15. Frederiks, P.J.M., Weide, T.P.v.d.: Information Modeling: the process and the required competencies of its participants. Data & Knowledge Engineering **58**(1), 4–20 (July 2006)

16. Gilby, T.: St. Thomas Aquinas Philosophical Texts. Oxford University Press (1951)
17. Gosala, B., et al.: Automatic classification of UML class diagrams using deep learning technique: Convolutional neural network. Applied Sciences **11**(9), 4267 (2021)
18. Guizzardi, G., Prince Sale, T.: "As Simple as Possible but not Simpler": Towards an Ontology Model Canvas. In: Borgo, S., Kutz, O., Loebe, F., Neuhaus, F. (eds.) Proceedings of the Joint Ontology Workshops 2017 – Episode 3: The Tyrolean Autumn of Ontology, Bozen-Bolzano, Italy, 2017 (2017)
19. Hake, P., Fettke, P., Loos, P.: Automatic pattern mining in repositories of graph-based process models. In: Multikonferenz Wirtschaftsinf. pp. 1143–1154 (2016)
20. Haugeland, J.: Artificial Intelligence: The Very Idea. Massachusetts Institute of Technology, USA (1985)
21. Heinrich, B., Klier, M., Zimmermann, S.: Automated planning of process models: Design of a novel approach to construct exclusive choices. Decision support systems **78**, 1–14 (2015)
22. Hjaltason, J., Samúelsson, I.: Automatic classification of UML Class diagrams through image feature extraction and machine learning (2015)
23. Hoppenbrouwers, S., Wilmont, I.: Focused Conceptualisation: Framing Questioning and Answering in Model-Oriented Dialogue Games. In: The Practice of Enterprise Modeling – Third IFIP WG 8.1 Working Conference, PoEM 2010, Delft, The Netherlands, 2010. Proceedings. LNBIP, vol. 68, pp. 190–204. Springer (2010)
24. Hoppenbrouwers, S.J.B.A., Lindeman, L., Proper, H.A.: Capturing modeling processes – towards the modial modeling laboratory. In: Meersman et al. [34], pp. 1242–1252
25. Hoppenbrouwers, S.J.B.A., Proper, H.A., Weide, T.P.v.d.: A fundamental view on the process of conceptual modeling. In: Delcambre, L., Kop, C., Mayr, H.C., Mylopoulos, J., Pastor, O. (eds.) Conceptual Modeling – ER 2005, 24th International Conference on Conceptual Modeling, Klagenfurt, Austria, October 24-28, 2005, Proceedings. LNCS, vol. 3716, pp. 128–143. Springer (June 2005)
26. Hoppenbrouwers, S.J.B.A., Proper, H.A., Weide, T.P.v.d.: Towards explicit strategies for modeling. In: Halpin, T.A., Siau, K., Krogstie, J. (eds.) Proceedings of the 10th Workshop on Evaluating Modeling Methods for Systems Analysis and Design (EMMSAD'05), held in conjunction with the 17th Conference on Advanced Information Systems (CAiSE'05). pp. 485–492 (2005)
27. Hoppenbrouwers, S.J.B.A., Rouwette, E.A.J.A.: A Dialogue Game for Analysing Group Model Building: Framing Collaborative Modelling and its Facilitation. International Journal of Organisational Design and Engineering (IJODE) **2**(1), 19–40 (2012)
28. Hotelling, H.: Analysis of a complex of statistical variables into principal components. Journal of educational psychology **24**(6), 417 (1933)
29. Jones, D., Snider, S., Nassehi, A., Yon, J., Hicks, B.: Characterizing the Digital Twin: A systematic literature review. CIRP Journal of Manufacturing Science and Technology **29**, 36–52 (2020)
30. Kelsen, P., Ma, Q., Glodt, C.: A lightweight modeling approach based on functional decomposition. Journal of Object Technology **19**(2), 15:1–22 (Jul 2020), the 16th European Conference on Modelling Foundations and Applications (ECMFA 2020)
31. Lakoff, G.: Women, Fire, and Dangerous Things: What Categories Reveal About the Mind. University of Chicago Press, Chicago, Illinois (1997)
32. Lakoff, G., Johnson, M.: Metaphors We Live By. University of Chicago Press, Chicago, Illinois (2003)
33. Lloyd, S.: Least squares quantization in pcm. IEEE transactions on information theory **28**(2), 129–137 (1982)

34. Meersman, R., Tari, Z., Herrero, P. (eds.): On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops, OTM Confederated International Workshops and Posters, AWeSOMe, CAMS, COMINF, IS, KSinBIT, MIOS-CIAO, MONET, OnToContent, ORM, PerSys, OTM Academy Doctoral Consortium, RDDS, SWWS, and SeBGIS 2006, Montpellier, France, October 29 – November 3, 2006. Proceedings, Part II, LNCS, vol. 4278. Springer (October/November 2006)
35. Mussbacher, G., Combemale, B., Kienzle, J., Abrahão, S., Ali, H., Bencomo, N., Búr, M., Burgueño, L., Engels, G., Jeanjean, P., Jézéquel, J., Kühn, T., Mosser, S., Sahraoui, H.A., Syriani, E., Varró, D., Weyssow, M.: Opportunities in intelligent modeling assistance. Software and Systems Modeling **19**(5), 1045–1053 (2020)
36. Proper, H.A., Bjeković, M.: Fundamental challenges in systems modelling. EMISA Forum **39**(1), 13–28 (2019)
37. Proper, H.A., Bjeković, M., Gils, B.v., Hoppenbrouwers, S.J.B.A.: Towards a Multi-Stage Strategy to Teach Enterprise Modelling. In: Aveiro, D., Guizzardi, G., Guerreiro, S., Guédria, W. (eds.) Advances in Enterprise Engineering XII – 8th Enterprise Engineering Working Conference, EEWC 2018, Luxembourg, May 28 – June 1, 2018, Proceedings. LNBIP, vol. 334, pp. 181–202. Springer (2018)
38. Proper, H.A., Guizzardi, G.: On Domain Modelling and Requisite Variety – Current state of an ongoing journey. In: Grabis, J., Bork, D. (eds.) The Practice of Enterprise Modeling. PoEM 2020. LNBIP, vol. 400, pp. 186–196. Springer, Riga, Latvia (November 2020)
39. Proper, H.A., Guizzardi, G.: On Domain Conceptualization. In: Aveiro, D., Guizzardi, G., Pergl, R., Proper, H.A. (eds.) Advances in Enterprise Engineering XIV – 10th Enterprise Engineering Working Conference, EEWC 2020, Bozen-Bolzano, Italy, September 28, October 19, and 2020, Revised Selected Papers. LNBIP, vol. 411, pp. 49–69. Springer (2021)
40. Proper, H.A., Weide, T.P.v.d.: Modelling as selection of interpretation. In: Mayr, H.C., Breu, H. (eds.) Modellierung 2006, 22.-24. März 2006, Innsbruck, Tirol, Austria, Proceedings. LNI, vol. P82, pp. 223–232. Gesellschaft für Informatik, Bonn, Germany (March 2006), `https://tinyurl.com/y3h4uas9`
41. Riefer, M., Ternis, S.F., Thaler, T.: Mining process models from natural language text: A state-of-the-art analysis. Multikonferenz Wirtschaftsinformatik (MKWI-16), March pp. 9–11 (2016)
42. Saini, R., Mussbacher, G., Guo, J.L.C., Kienzle, J.: DoMoBOT: a bot for automated and interactive domain modelling. In: MODELS '20: ACM/IEEE 23rd Int. Conf. on Model Driven Engineering Languages and Systems, Canada, 18-23 October, 2020, Companion Proceedings. pp. 45:1–45:10. ACM (2020)
43. Sandkuhl, K., Fill, H.G., Hoppenbrouwers, S.J.B.A., Krogstie, J., Matthes, F., Opdahl, A.L., Schwabe, G., Uludag, Ö., Winter, R.: From Expert Discipline to Common Practice: A Vision and Research Agenda for Extending the Reach of Enterprise Modeling. Business & Information Systems Engineering **60**(1), 69—80 (2018)
44. Sherstinsky, A.: Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena **404** (2020)
45. Van Otterlo, M., Wiering, M.: Reinforcement learning and markov decision processes. In: Reinforcement learning, pp. 3–42. Springer (2012)
46. Zapp, M., Fettke, P., Loos, P.: Towards a software prototype supporting automatic recognition of sketched business process models. Wirtschaftsinformatik und Angewandte Informatik (2017)
47. Zarwin, Z., Bjeković, M., Favre, J.M., Sottet, J.S., Proper, H.A.: Natural modelling. Journal Of Object Technology **13**(3), 4: 1–36 (July 2014)