Towards Holistic Conceptual Modeling Education: Experience Report on Course Redesigns

Robert Andrei Buchmann*, Cristina-Claudia Osman, Ana-Maria Ghiran, Damaris-Naomi Dolha and Anca Moldovan

Babeș-Bolyai University, Teodor Mihali 58-60, Cluj-Napoca 400591, Romania

Abstract

This is an experience report on the course re-design efforts and learned lessons from the authors' host institution, regarding Conceptual Modeling topics. The report is centered on a Systems Analysis & Design course for the Business Informatics bachelor studies program, accredited in the domain of Business Administration. The course establishes a core modeling-based body of knowledge for business analysts and system engineers, which later diverges into related disciplines at master and doctoral levels. The effort started from a legacy UML and requirements engineering course that used to generate what students perceived as a "Diagrammer" skill profile, and the restrictive conclusion that conceptual modeling methods belong heterogeneously to other disciplines, rather than being knowledge capture and analysis enablers offered by a standalone discipline. Based on experience exchanges within the modeling-focused OMILAB community of practice, and inspired by agendas and re-framings advocated in recent scientific literature on Conceptual Modeling education, the authors have pursued the initiative to redefine the curricular SA&D offer as a layered approach. The new approach balances a diversity of tool practice, multi-perspective conceptualization and knowledge work across multiple abstraction layers and standards. To give a constructivist flavor to the learning path, this is synchronized with the documentation requirements for bachelor theses projects. The effects of this re-design show primarily a shift from the "graphical modeling" focus, resulting in a steady stream of publications now realized by students, emerging doctoral projects aiming for Conceptual Modeling contributions (not only tool usage), and benefits pertaining to a better coverage of Bloom's taxonomy regarding Conceptual Modeling learning objectives and competence.

Keywords

Conceptual Modeling education, Bloom's taxonomy, Model Queries, BPMN, UML, Archimate

1. Motivation

This is an experience report of the on-going effort, over the last decade, to redesign the Conceptual Modeling (CM) educational offer for the Business Informatics study program in the authors' host institution. It is a sister report to [1], which focused on similar parallel efforts (from partly the same team) regarding the teaching of Knowledge Graphs and related semantic technologies. That study mentioned several points of convergence between non-diagrammatic conceptual modeling (knowledge graphs and ontology engineering) and diagrammatic conceptual modeling, which are achieved at master and doctoral programs level by building on the foundation hereby presented.

Therefore, in this paper we zoom in on a bachelor-level Systems Analysis & Design (SA&D) course, a curricular slot recommended by the joint ACM/AIS competence models [2] that has been traditionally covered by a legacy course on UML and requirements engineering. Over the recent years, we have upgraded this through several course re-designs, for a number of reasons:

- partly motivated by recent academic agendas and recommendations to establish CM as *a standalone scientific discipline* [3], with a constructivist teaching approach indicated as dominant in a

ER2025: Companion Proceedings of the 44th International Conference on Conceptual Modeling: Industrial Track, ER Forum, 8th SCME, Doctoral Consortium, Tutorials, Project Exhibitions, Posters and Demos, October 20-23, 2025, Poitiers, France

*Corresponding author.

robert.buchmann@ubbcluj.ro (R. A. Buchmann); cristina.osman@econ.ubbcluj.ro (C. Osman); anamaria.ghiran@econ.ubbcluj.ro (A. Ghiran); damaris.dolha@econ.ubbcluj.ro (D. Dolha); anca.moldovan@econ.ubbcluj.ro (A. Moldovan)

0000-0002-7385-1610 (R. A. Buchmann); 0000-0002-9706-2915 (C. Osman); 0000-0001-7890-9386 (A. Ghiran); 0009-0001-7695-4383 (D. Dolha); 0009-0005-7273-1325 (A. Moldovan)

general landscape of *highly fragmented CM education* [4];

- motivated by successful course re-designs reported by universities with strong CM tradition [5] and the emergence of CM educational frameworks based on Bloom's taxonomy [6];
- inspired by the new CM research agenda bringing forth the *mediation role* of CM in digital worlds [7] and digital twins [8], i.e. shifting the main purpose of CM from visual representation to multi-modal knowledge structuring and mediation in model-driven environments;
- informed by the experience exchange network of the OMILAB Community of Practice[1], where we have been part as active members and occasional organizers of CM experience exchange workshops - see the OMILAB-KNOW[2] and PrOSE workshops series[3];
- driven by our own Design Science framing of CM education as formulated in [9].

The remainder of the paper is structured as follows: in Section 2 the course redesign treatment will be framed as a Design Science Research (DSR) problem, followed by insight into the problem context in Section 3. Section 4 describes the evolution stages of this curricular re-design from the legacy stage to the current one, further detailing rationale and learning outcomes based on a revised Bloom taxonomy. Section 5 enumerates the teaching tools and key enablers. Section 6 comments on related work and Section 7 provides theoretical connections to educational theories. The paper concludes with a SWOT evaluation summarizing some success indicators and recommendations.

2. Design Problem and Requirements

Design Science Research (DSR) being an appropriate frame for proposing prescriptive treatments to contextualized problems, we remain anchored with this proposal in the position that was formulated in [9]. However, we update the generic "design problem" formulated there and specialize it for the particular SA&D course under scrutiny in this paper, structuring its statement according to the traditional DSR template recommended by [10]:

Enhance the **CM curricular offer in the host institution for the field of Business Information Systems** *(problem context)*
… by treating it with **a course redesign that can position CM as a standalone discipline offering diverse viewpoints and traceability analysis paths** *(treatment)*
… to facilitate the emerging **mediation and viewpoint integration roles of CM** *(treatment service/requirements)*
… in order to support system analysts **who need multi-perspective design traceability and semantic competency going beyond diagramming/visualization – i.e. the legacy focus of the preexisting course design** *(stakeholder goals)*

This was further drilled down into epistemic goals that drive this effort, derived from requirements from upper levels courses (master and PhD) that rely on certain CM foundations:

R1. To reveal a diversity of modeling languages in relation to diversifying purposes, thus showcasing how CM offers instruments to various Business Informatics disciplines - some leaning on business, others on information technology, or alignining the two. We frame this diversity of purposes according to the Purpose-Specificity framework [11] and discuss both occasional overlapping and semantic divergence of several modeling languages/standards, finally acknowledging their inherently limited scopes, and also their complementary competence reflected in metamodels. This serves to detach CM applicability from the legacy software engineering goals (already supported by other courses, e.g. Databases and Object-oriented Programming), i.e. moving beyond the UML standard towards a diversity of modeling cases that are relevant for our Business Administration school: business processes management (BPM), case management, decision management, enterprise architecture management (EAM).

[1]https://www.omilab.org/
[2]https://www.omilab.org/activities/events/bir2024-ws/
[3]https://austria.omilab.org/psm/content/prose2019/info

R2. To switch the dominant "Diagrammer" skill profile - inherited from the legacy course - to a novel knowledge work skill profile, previously advocated under the label "Digital Innovator" in the educational agenda of the OMILAB network [12]; or, as "Complexity Manager" when the goal is to drill down complex systems or project them across viewpoints. The master programs further carry over such CM foundations, towards model-driven engineering (MDE), knowledge graphs, low-code and other system engineering concerns associated with the competence profile advocated in [12].

R3. To switch the legacy perception - that the main purpose of modeling is to obtain some diagrams - towards the understanding that equally relevant is the modeling cognition, i.e. the underlying effort and side-effects - situational comprehension, sense-making, gap filling - that a system analyst goes through while creating diagrams. This effort includes for example identifying decisions and corner cases in case management, identifying dependencies towards architectural completion, comparing alternate chronologies in a process design, ultimately ensuring the right level of detail becomes available for subsequent retrieval and traceability, in external systems relying on the model contents. This also gives occasion to discuss the ChatGPT-driven irrelevance of output-based student evaluation (even encouraging assessment automation in past literature [13]). Instead, we revert back to a full Bloom-based evaluation effort across several layers of the taxonomy or its variants [14]. After all, "understanding" cannot be detached from the earliest definitions of CM [15] and the aforementioned CM role shifting should not equate with knowledge work de-skilling through AI-delegated modeling. This is also in line with competence models discussed in the literature for graphical modeling [16], but also goes a step beyond, as the next requirement explicates:

R4. The above mentioned goal of "retrieval and traceability" (in external model-based systems) implies that model querying demonstrations are provided as early as possible to students - e.g. starting with process querying methods [17] as an intuitive introductory exemplification, but not limited to that. Model queries can be turned into a flavor of "competency questions" from knowledge engineering [18] – but confined to the competence boundaries of each diagram type or standard, according to their metamodels. Such an approach transcends established CM educational competence models such as CMGM (Competence Model for Graphical Modeling) [16] - that put at the forefront "graphical modeling" assuming the Diagrammer profile and a visual communication focus.

To satisfy all the above requirements, an overarching line of argumentation must encompass the course progression and learning objectives, summarized in several stages depicted in Figure 1. This progression starts from highlighting the diversity of CM methods and standards, to being able to choose between them in relation to purpose and desired competence. On the right side of the figure, there is a suggestion of how this flow advances to the master program. In the master program, we can directly step into DSML (domain-specific modeling language) engineering as a means of expanding the competence/scope of any modeling standard, and therefore the range of competency questions that diagrams can answer. This is based on educational tooling that easily demonstrate the focus on model contents semantics and metadata, and not only visualization. Exemplary topics that take this idea further and demonstrate diverse applicability are studied in the English-language master program of Business Modeling and Distributed Computing[4]:

- Process queries, simulation and mining (in BPM context)
- Enterprise Architecture reporting (in EAM context)
- Low-code (in Model-driven engineering and Internet-of-Things context)
- Digital Twins (in Model-driven Knowledge Graphs context, primarily Knowledge Graphs derived from BPMN models as advocated in recent experimental demonstrators [19])

These topics have been picked not only because of a desire to balance business and informatics topics, but also due to their reliance on relational traceability and model navigability, thus showcasing a diverse applicability layer on top of the CM foundations established by the course hereby discussed. We found it critical to introduce **model queries** from the earliest meetings (briefly showing demonstrators, e.g.

[4]https://econ.ubbcluj.ro/programe/bmdc/

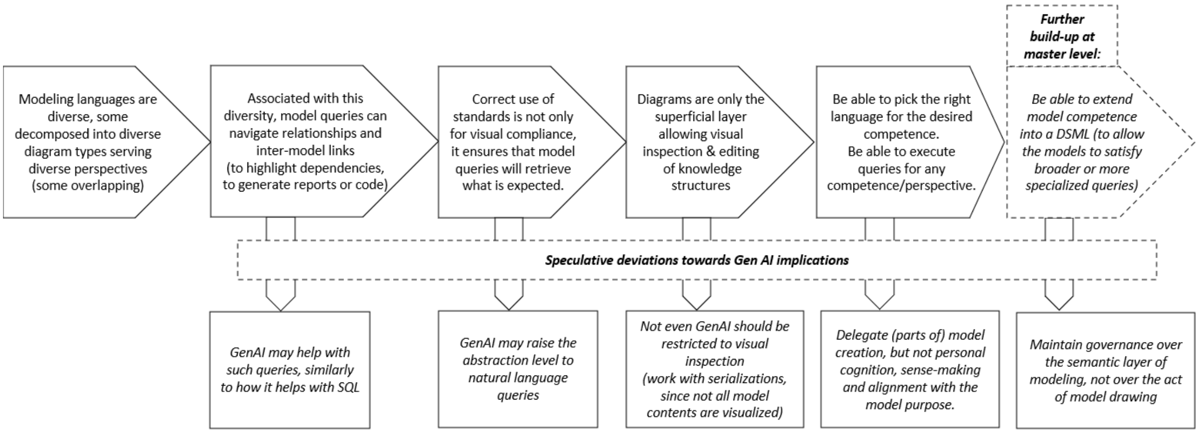

Figure 1: Line of argumentation overarching the course progression design, to shift the CM role from Representation to semantic Mediation.

report generation) as this is perhaps the one unexpected feature that easily catches attention, sparks curiosity and stimulates lateral thinking for those students stepping into the course with the legacy mindset of the Diagrammer profile. The legacy profile is sometimes inherited from past generations, or from Web presentations that superficially introduce CM as a "drawing" activity, thus maintaining for novices a disconnect from subsequent model-based activities and implications. Consequently, our choice of educational tooling (see Section 5) aims to avoid simple diagramming canvases where visual sketching and inspection are the only tasks supported.

Some points of discourse deviation are also shown as downwards arrows in Figure 1, where the potential involvement of **Generative AI in CM activities** is suggested. This has become a natural curiosity for students, and the CM-GenAI interplay is increasingly prevalent in literature – from investigating how GenAI can serve the Business Process Management lifecycle [20, 21], to more diversified CM activities at modeling and metamodeling levels [22, 23, 24]. We also refer to our own work on assessing differences between large language models' ability to interpret diverse model representations – XML, RDF, images [25]. This enables us to advocate once again for the importance of querying over model serializations as carrying supplementary information - not always available to visual inspection by computer vision. While the course does not currently involve explicit tasks involving GenAI-CM interaction, we added them to the figure as suggestions of possible divergence points, if CM capabilities in GenAI products will evolve into sufficiently reliable ones (or if free educational CM tools will start incorporating GenAI features).

3. Problem Context

CM education has always faced dilemmas in the search for an ideal balance between practitioner-oriented relevance and formal grounding, between tooling and standardized principles. Challenges can also be case-specific, such as in the regional context where this report originates – an IT outsourcing hub that advertises cost savings, traditionally considering CM efforts to be a costly overhead of little pragmatic relevance, for a number of diverse reasons:

- Design decisions are often taken for granted or inherited from legacy systems being maintained by the outsourced service provider. Long-term loyal partnerships lead to intimate understanding of the client-side business processes and legacy architecture – already available in past documentation, or shared through knowledge management approaches (wikis, Confluence etc.);
- IT services tend to be specialized and sometimes tackle only small parts of the software development lifecycle – software testing, front-end "face lifting", DevOps etc. Workers involved in such isolated concerns are often not interested, or do not have access to a system-wide view of the

enterprise system they contribute to. The argument of "agility" is occasionally brought forth to justify this, even when it leads to lack of awareness of certain dependencies;
- Business/process analysts have been absent from software projects for many years, until some customers or certification requirements imposed their presence on the outsourced side. A typical example is the sudden success of RPA (Robotic Process Automation), where outsourcing providers quickly set up tool-specific (UiPath, Power Automate) developer teams deploying RPA implementations in the absence of a Business Process Management culture. This has lead to widespread improvisation of process description methods neglecting BPMN and ad-hoc process analysis methods based on spreadsheets and visual inspection.

On the other hand, recent crises waves in the IT industry – from the pandemic to several stages of automation – revealed new distinctions in capabilities where CM can play a role, even if not explicitly labelled as CM (practical work calls it "process mapping", "low-code development", "visual maps" or, simply, "diagramming" or "design"):

- Distinctions between reactive/reflexive companies, preoccupied strictly with identifying opportunities to offer highly specialized services and human resources, as opposed to companies developing their own products, even meta-products whose design decisions must be tailored to diverse business processes and domain specificities;
- Distinctions between companies that lost their outsourcing partnerships (because of e.g. backshoring in the context of RPA and AI), as opposed to companies that managed to turn themselves into automation providers, including associated consultancy "centers of excellence";
- Some companies indirectly acquired CM talent and capabilities after facing impositions from customers or projects to demonstrate availability of knowledge work roles (process analysts, knowledge managers, enterprise architects) or certifications (ITIL, TOGAF etc.);
- Some companies shifted from using commercial low-code platforms to developing their own low-code environments to improve productivity and expand development capabilities. This took them inevitably on a model-driven engineering path.

On the academic side, the curricular offer accommodated such shifts over time in a mostly ad-hoc manner, reacting to hypes, adopting tool-centric tutorials and not really pursuing an explicit CM competence model/skill profile. Some disciplines always incorporated CM tooling (ER in Databases, class diagrams in Object-oriented Programming), while others followed industry trends (e.g. RPA as a tool-centric tutorial in the absence of any Business Process Management or Enterprise Architecture contextualization). CM methods and tools have spread between disciplines in a highly fragmented manner, which is in line with the conclusion of the literature survey in [4] about a similar fragmentation of model types used in the CM education scientific literature.

As a group of educators responsible with managing a stream of courses – starting from SA&D at bachelor level (the focus of this paper) and later diverging at master level into BPM, EAM, RPA, MDE and low-code IoT – we engaged in a longitudinal curricular redesign and streamlining that positions CM as a standalone discipline with a rich offer of tooling and methodological enablers to a diversity of disciplines in Business, Informatics, or their convergence.

4. Evolution Stages and Learning Objectives

After first introducing, for a research-oriented master program (around 2015-2017) several teaching artifacts involving DSMLs and experimental interplay between knowledge graphs and diagrammatic modeling (see Section 6 and the sister report in [1] about such priors), it was concluded that insufficient knowledge and skillsets are carried over from the bachelor program. We identified a gap between the bachelor-level legacy courses involving CM and the expectations to successfully involve CM later, in educational tasks on MDE, process automation, EAM, low-code IoT and knowledge graphs - all these manifesting as emergent topics at the time. To support this emergence, we proposed to introduce CM as a multi-perspective conceptualization and knowledge refinement approach, thus engaging in several evolution stages for the bachelor-level course:

The Legacy Stage: Stable curricular content has been maintained for close to a decade before we started getting involved in this course. It used to be based on a selection of core UML diagrams (use case, activity, state, class), complemented by data modeling (ER), and book-ended by informal visualizations (e.g. the Fishbone diagram) and software project management considerations on Waterfall and various Agile flavors. This was reasonably aligned with the joint ACM/AIS recommendations for SA&D competences [2], and also with competence models for graphical modeling [16], but it was inherited from a software engineering focus in the early days of the program. For a Business Informatics program in a Business Administration faculty, it largely missed standards and methods for business analysis, enterprise architecting and business-IT alignment. Moreover, over the years some redundancy emerged compared to specialized courses on Databases (ER modeling), Object-Oriented Programming (UML class diagrams) and Software Development Life Cycles. We chose to minimize redundancy by maintaining only brief references to those (all of them taking place earlier in the curriculum), and thus to make room for expanding the perception on CM as a unified and diversified discipline.

Stage 2. Diversification of Conceptualization: The new approach allows to navigate several standards and multiple tools, enabling discussion on model competence overlapping and complementarities. Subtly, this set the grounds for later introducing model queries to highlight the value of formal modeling for machine-readable enterprise knowledge structuring. Parts of UML and data modeling, already covered by prior disciplines, are only briefly revisited; instead, the focus is shifted to business process modeling and architecting, designing systems primarily around **chronologies** (workflows, data flows) and **architectural dependencies**, which tend to become prominent requirements in the local industry, especially since the uptake of Robotic Process Automation.

We thus introduced the BPMN ecosystem, including coverage of complements such as CMMN, DMN, DFD. Further down the road, workflows are contextualized by enterprise architecting, where traditional UML diagrams (use case, deployment, components) are compared to the more integrated approach of Archimate. This helps stress the multi-viewpoint modeling needed to manage architectural complexity and enterprise layering. Archimate is not presented in full due to time constraints, so we focus on (a) comparing its core layers to BPMN and UML, and on (b) wrapping around those core layers the Physical deployment of a business, and the Motivation layer (later, at master level, Archimate is picked up in the context of TOGAF[5] and the remaining layers are added). In close relation to the Motivation layer of Archimate, informal diagrams are introduced as possible input for motivation modeling. These informal visual templates include Fishbone, SWOT analysis and Business Model Canvas[6] to capture high-level business model summaries as a starting point for architecting an enterprise or its information systems.

Stage 3. Moving Beyond Diagramming: One major acceptance obstacle in the UML-focused legacy course, also partly manifesting in the updated version described as Stage 2, was a dominant perception that the course is about various kinds of "drawings" contributing to a *Diagrammer skill profile*. In students' feedback, this profile was associated with several demotivating assumptions that needed to be addressed:

- that the Diagrammer work is mostly motivated by a pedantry in complying with graphical standards, when informal diagramming (Miro, Draw.io etc.) is just as good for intuitive communication – local industry practice being actually dominated by informal approaches rather than by formal standards, under the pretext of Agile development moving away from the "UML days";
- that the Diagrammer goal is to obtain some diagrams, and this effort can now be delegated to Generative AI; moreover, large language models allow for the Diagrammer to be replaced entirely by natural language "storytelling" as means of both knowledge acquisition and representation, i.e. reducing diagrams to a historic compromise.

Several countermeasures were necessary to push through such assumptions and defuse the legacy Diagrammer competence profile. This is based on stimulating a constructivist CM education approach

[5]https://www.opengroup.org/togaf
[6]https://www.strategyzer.com/library/the-business-model-canvas

by showcasing from the earliest courses demonstrators or cases where those assumptions do not hold, where multi-perspective enterprise modeling and systems engineering gain value:

- The legacy Diagrammer competence profile, based on small isolated diagramming exercises that mostly stress syntactic correctness, used to render opaque all **CM cognitive efforts** – i.e. knowledge acquisition, sense-making, drill down and gap-filling. These are the basis of understanding a system-under-study and documenting its As-Is vs. To-Be vision gaps. Diagrams are presented as a proof that the cognitive effort took place, that the situational understanding was achieved – thus delegating the effort of "drawing", is not the same as delegating situational understanding, identification of types of things and dependencies manifesting between them. Students are tasked not only with isolated diagramming exercises, but with a complex software project where their software development efforts run in parallel with enterprise modeling, business analysis and business-IT alignment based on CM methods. This is enforced through Bachelor thesis guidelines, as the final thesis project is streamlined with the SA&D course to allow for complexity management concerns to arise, finally justifying the aforementioned cognitive efforts - as a learning experience clearly distinct from graphical content creation.
- As a communication device, students are shown as early as possible the empirical effectiveness of **visual inspection of a complex diagram**, compared to the same situation expressed via storytelling. It is advisable to organize exercises where question answering accuracy from students vary between having to inspect a text and a diagram of the same situations, especially when the questions involve multi-hop navigation of connectors/dependencies, more difficult to grasp from pages of linear text;
- Afterwards, this visual inspection is replaced by **model queries** to highlight how the apparent "visual pedantry" is actually an effort of reducing ambiguity for both humans and machines, i.e. allowing queries to deterministically distinguish and retrieve desired element types in support of automated reporting or model-driven processing. Most demonstrated model queries are based on the standard XML serializations available as export formats in all modeling tools involved – i.e. XPath[7] is shown as the most general retrieval approach applicable across all studied standards and tools, while also mentioning some tool-specific ones (e.g. the ADOxx query language[8] applicable to BPMN, UML, DMN, UML in the Bee-Up tool[9]) and others reported in the scientific literature [26]. XPath expertise is already available from other courses - it is involved in software testing, Web scraping or XML content management, a convenient shortcut to quickly highlight the machine-readability of diagrammatic content - a CM aspect that is thus pushed in front of the traditional visual concerns.

At this stage, different model competencies, each illustrated by key concepts - chronology (of actions, events), data (structuring and requirements), architectural dependencies – are navigated by queries tailored for all modeling standards, which helps establish quickly that diagramming is only a superficial visual layer wrapping around the mediator role of CM, with its knowledge structuring and traceability concerns. Some examples of queries that can be easily associated with pragmatic use cases are:

– *a list of BPMN user tasks to be turned into Jira tasks*
– *a list of DMN data inputs to be turned into data requirements for a decisional API*
– *a list of Archimate elements that influence a specific Motivation layer object.*
– generalized template applicable across standards: *a list of elements of a specific type, involved in a specific relationship or relationship chain.*

Recurring query patterns are discussed - first as natural language questions that students must be able to formulate in order to demonstrate their understanding of the scope/competence of each type

[7]https://www.w3.org/TR/xpath-31/
[8]https://www.adoxx.org/documentation/75_adoxx_development_languages/01_AQL.html
[9]https://bee-up.omilab.org/activities/bee-up/

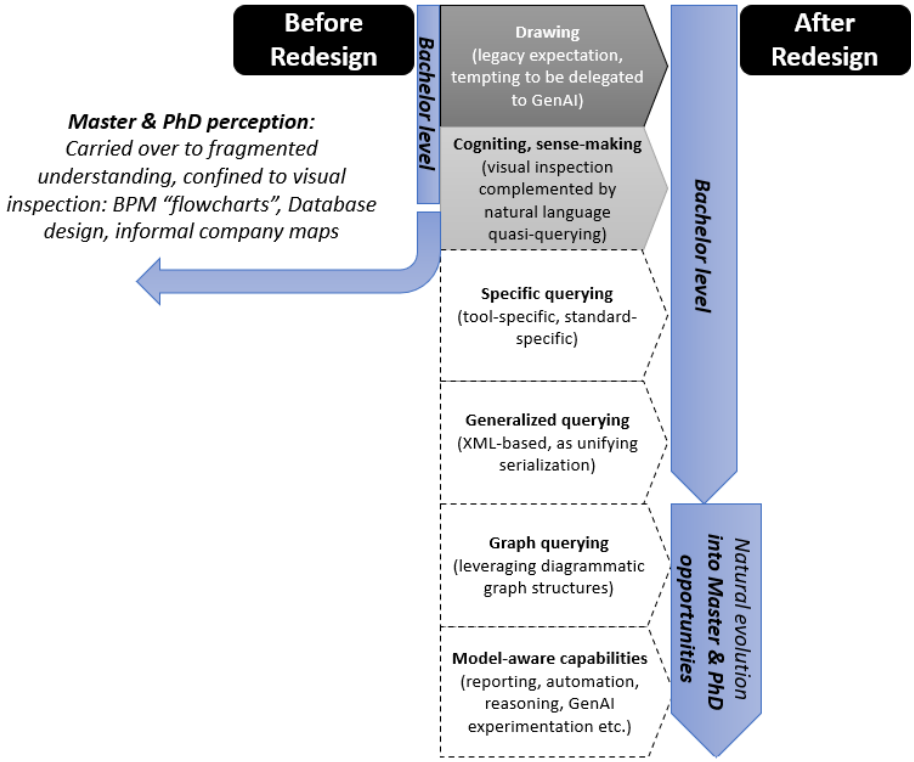

Figure 2: Evolving model queries to expand CM understanding from drawing to model-driven engineering.

of model. An early observed obstacle is the traditional understanding of queries (as "data queries, e.g. SQL) that comes with a tendency to formulate questions about potential model traces or execution-time instances (who are the employees that executed task X?), instead of querying visible model contents (to what participant lane/pool does task X belong?). After this distinction is clarified, the question patterns can be gradually translated into executable queries while diversifying the technical means to execute them - we insist on XPath queries over XML serializations at bachelor level, after showcasing a few tool-specific features. However, as shown in Figure 2, later at master level this advances to more specialized query methods (RDF-based queries as allowed in the OMILAB ecosystem toolkits [12]) and to the retrieval of model contents in model-driven software demonstrators. To foreshadow master studies, a few model-based demonstrators are briefly shown even at bachelor level - for low-code development (e.g. BPMN-based Tasky[10]), architecture analysis and traceability reporting (ADOIT[11]), process analysis and simulation. Also suggested through Figure 2 is the fact that, compared to this shift of focus, the legacy perception used to generate a fragmented perception dominantly connected to CM as a general purpose visualization activity.

Stage 4. Enforcing CM Constructivism: Bachelor Thesis guidelines were developed to enforce the reduction of narrative text in theses (lately susceptible to GenAI contamination) and to stimulate a switch towards presenting and arguing for design decisions and business rationale with the help of CM tools and standards. This varies with the type of thesis (also allowing combinations):

Type 1. Software development projects (dominant, about 80% of bachelor theses) must be framed in the written thesis by a business model and enterprise architecture where the developed software could play a role. Such documentation starts from high-level informal considerations - Fishbone causality and unstructured mind maps ideation -, refined into Archimate Motivation and Business Model Canvas, then turned into business processes and finally going through granularity levels of system architecture. Students have the choice to pick their favourite architecting language (UML, Archimate, DFD) and business process perspective (i.e. between strongly structured workflows and case

[10]https://admin.tasky.cloud/
[11]https://adoit-ce.boc-cloud.com/cp/

management, possibly involving DMN if business rules play a role). Contrasting between the As-Is vs. To-Be models is important to highlight the business transformation that the proposed software artifact brings.

Type 2. Case studies from the students' workplaces, typically pertaining to the adoption of new work procedures or enterprise systems is second category of theses; here, CM serves mostly for As-Is vs. To-Be comparisons and some characterization of existing software. This type of thesis is a minority, although not rare (15-20%).

Type 3. Scientific research makes less use of CM methods - mostly to reflect empirical data processing concerns (ata processing pipelines and data structures), unless CM itself is at the core of this research. This type of thesis is, however, a rarity at bachelor level, as it requires scientific methods only introduced in master studies.

Table 1 summarizes learning objectives that are the basis for examination in relation to the revised Bloom taxonomy of learning levels proposed in [6, 14]. Each row lists outcomes for both the bachelor-level SA&D course and its immediate follow-up at master level (focusing on DSML development before diverging to various model-driven engineering flavors).

Table 1
Outcomes according to the Bloom (revised) taxonomy

Taxonomy level	Bachelor-level outcomes	Master-level outcomes (around DSML development)
Remember	The ability to recall and give informal definitions on concepts present in the investigated standards – e.g. BPMN gateway, CMMN case, DFD entity (e.g. vs ER entity), Archimate business function etc.	The ability to recall and give informal definitions on the building blocks of a modeling method, according to established DSML engineering methodologies e.g. Agile Modeling Method Engineering phases [27].
Understand	The ability to describe in natural language a situation depicted in diagrams with non-explicit labeling, i.e. enforcing the recognition of model elements, rather than extrapolation from textual labels.	The ability to describe in natural language a metamodel depicted as a class diagram, and to instantiate a case as a mock diagram conforming that metamodel.
Apply	The ability to "translate" a piece of natural text into a diagram of a specified type, preserving as much as possible of the details provided in the textual description.	The ability to draw the domain-specific metamodel capturing all relationships and types involved in a natural language description of a situation.
Analyze	The ability to assess whether a competence question can be satisfied by the contents of a diagrammatic model, with respect to the scope of the modeling standard being used. The ability to execute model queries in order to produce a simple report listing model elements that satisfy certain restrictions.	The ability to assess whether a competence question can find its answer in models created accord-ing to a given domain-specific metamodel (given a legend a mock symbols).
Evaluate	The ability to find mistakes in models, and to distinguish between syntactic mistakes, missing information or mistakes relative to the situation being represented.	The ability to assess whether a domain-specific diagram deviates from a metamodel it is supposed to be compliant with.
Create	The ability to "translate" a large-scale situation (the Bachelor thesis business case) into a collection of adequate multi-perspective diagrams of different types, depicting different aspects of a situation/system - using the preferred tooling, while ensuring that software implementation is in synch with the diagrammatic design decisions.	The ability to implement a novel modeling tool that deploys a metamodel designed by students. The ability to use it to describe instance situations. The ability to build a front-end that demonstrates the retrieval of model elements at run-time.

5. Key Technological Enablers

The tooling involved in the new SA&D course design is an assortment of diverse educational tools available to students:

- On one hand, we employ SAP Signavio[12] (for BPMN, CMMN, DMN and Archimate), Archi[13] (for Archimate) and Visual Paradigm[14] (mainly for DFD but it also serves many other languages);
- In parallel, we employ toolkits from OMILAB's Digital Innovation environment [12], mainly Bee-Up[15] freely available to the OMILAB Community of Practice, which offers support for UML, BPMN, DMN, ER and others, while also offering a model query engine (based on ADOxx QL[16]) and more diverse exporting options (RDF export for all diagram types);

Between the dominant tasks - modeling, model analysis/querying - other tools may be shown as teasers for the master program: BPMN automation in Tasky Cloud[17], Archimate analysis reports in ADOIT[18], ADOxx[19] to expand modeling standards and querying possibilities towards DSMLs, UML class code generation in Bee-Up, DMN and Process simulation etc. However these are brief intermittent detours, as practical tasks remain focused on gaining familiarity with the diversity of modeling grammars and standards, understanding their specialized competence and overlaps, gaining hands-on experience with diverse tooling and being able to extract model information via queries. The rationale for the selection of tools, besides their open availability to students, is:

- to avoid creating the impression of a tool-centric tutorial (e.g. "this is the StarUML course", a perception detected in the legacy stage of the course);
- to enforce a clear distinction between a modeling language/standard and tooling, raising awareness of differences and overlaps;
- to allow any student to pick preferred tooling for their Bachelor theses;
- to avoid diagramming-only tools that are not able to demonstrate any model-based feature outside visual formatting (a critical point in defusing the Diagrammer focus).

6. Prior and Related Work

As background priors, we refer to some of our earlier reports that provided insights to teaching cases and courseware that was experimentally employed at master and doctoral level before taking over the bachelor-level content: the "cooking recipe modeling method" gives insight to the constituents of a minimalist rudimentary workflow language [28], an IoT-focused DSML [29] advocates for a "modeling at large" approach where modeling grammars, methods and tools are tailored to targeted competence queries within a Purpose-Specificity orthogonal space [11]. One key commonality derived from such exemplars is their reliance on model querying methods to demonstrate the diversity of purposes and specificities – from the built-in querying methods of ADOxx[20] to SPARQL on RDF graphs derived from a network of interconnected models, as facilitated by an ADOxx-specific export plug-in [30].

Those enablers and teaching strategies, however, required a foundation established in the bachelor level CM-focused course presented in this paper. Therefore, the course takes a generalized approach to model queries relying on the generalized availability of XML serializations, leveraging existing XPath skills gleaned from tasks in other courses. The ability to use queries to navigate the dependencies and

[12]https://academic.signavio.com/
[13]https://www.archimatetool.com/
[14]https://www.visual-paradigm.com/
[15]https://bee-up.omilab.org/activities/bee-up/
[16]https://www.adoxx.org/documentation/75_adoxx_development_languages/01_AQL.html
[17]https://admin.tasky.cloud/
[18]https://adoit-ce.boc-cloud.com/
[19]https://adoxx.org/
[20]https://www.adoxx.org/documentation/75_adoxx_development_languages/01_AQL.html

decompositions captured in models becomes thus a unification concern across all CM standards or tools, as a more powerful and deterministic model process approach compared to solely visual navigation. It is also a key message against perceiving CM as a fragmented visually-focused practice, reframing it as a knowledge refinement and retrieval practice.

For CM education research this proposal opened new possibilities for empirical studies - e.g. with Generative AI model queries across different model types, patterns and formats [25]) - that go beyond the traditional concerns of human-to-human communication via models, graphical modeling competence [16], visual comprehension [31], cognitive effectiveness [32], modeling styles [33, 34] or modeling errors [35].

Recent scientific agendas have also proposed a reframing of CM education as a standalone discipline offering technological and methodological enablers to other disciplines, as opposed to the traditional fragmentation. The extensive bibliometric analysis of [4] takes a holistic approach to survey all areas of CM education reports to reveal this fragmentation, while [3] focuses on software modeling but still identifies the characteristics of a "scientific discipline". The survey of [36] raised the specific point of design research as being insufficiently represented in comparison with empirical reports – i.e. to propose treatments and artifacts that can enhance CM education tasks, from learning to evaluation; it is a proposal in line with our agenda of formulating CM education itself as a "design problem" [9]. The mentioned prior works [28, 29] and others shared in recent OMILAB workshops [37] have been discussing mainly software artifacts, while in this paper we discuss a more strategic treatment that orchestrates a variety of tools and conceptual framings about the purpose and methods of CM.

7. Connections to Learning Theories

Constructivist learning [38] recommends that students should derive knowledge by their own construction effort rather than by direct assimilation of content or imitation tasks. The CM education research survey of [4] points to a dominance of constructivist approaches being reported by the literature. A much earlier epistemological recommendation was formulated in [39], placing CM as a core discipline of the Information Systems field and contrasting its priorities of "construction" against the dominant empirical interests in the field. This even included the construction of not yet existing realities (as opposed to representing empirically observed ones) which foreshadowed the CM agenda on the representation-to-mediation role shifting formulated by [7].

In our case, students end up performing a guided construction effort mainly for their bachelor thesis, where the diversity of modeling languages and tools are used in a coordinated manner to describe bachelor projects from multiple perspectives, both business and software-oriented. Most projects are software systems that must be not only developed but also contextualized by a (real or fictive) business model, business processes and enterprise architecture. The SA&D course runs in parallel with this construction effort by gradually introducing the different model types and competencies involved there. The course builds on several prior modeling skills presented in earlier disciplines in a fragmented manner (mainly data modeling presented in Databases courses as part of the relational DB design process) and on XPath as generic model navigation and retrieval means - to stress that models are not an end in themselves, but must interoperate in larger systems as semantically-prescribed mediators.

Through its ramifications to other already mentioned master program disciplines, the course also follows the spiraling strategy recommended by J. Bruner [40] - i.e. a recurring revisitation of the same topics throughout the curriculum. XPath queries later diversify into more specialized model query methods [17, 30] (see Figure 2), and the diagram types covered in this course become the basis for process automation, enterprise architecture reports and model-driven engineering of various other artifacts.

8. Concluding SWOT Analysis

Strengths. The main goal of this longitudinal course redesign effort is to enable, in the spirit of the humboldtian education model, a capability of producing CM scientific output authored by students and at the same time a constructivist approach to CM education. The students become able to create a model-based knowledge repository for their system engineering outcomes. This also makes them more prepared for junior research work in projects and for following a CM-centered PhD program – which only became possible recently; conference publications based on student theses or their research project work are now regular - we are tracking them at https://econ.ubbcluj.ro/omilab/publications.php, with typical venues being BIR, CAISE workshops, KES, ECIS, ISD, often in tool/forum sections due to the practical focus of student work. Institutional projects became possible, employing student talent now familiar with purpose-divergent CM methods, including DSMLs; one recent example of a successful institutional project involves a student-developed DSML for managing process design decisions in Robotic Process Automation projects [41].

Weaknesses. One major shortcoming for an SA&D course is the absence of SysML, which we postponed in order to wait for the new, radically re-designed version announced for 2025. By reducing current UML coverage, already partly represented in a parallel software engineering courses, we can bring in the new vision SysML that also shares our concern for models as logical knowledge structures not necessarily visualized diagrammatically.

Opportunities. We have yet to look at the emerging opportunities brought by new versions of SysML and Archimate, in relation to the noticeable interest in the convergence of modeling languages and RDF/OWL, see the knowledge graph taskforce recently established by OMG[21]. This has the potential to standardize efforts regarding ontological commitment [42], reasoning and semantic querying [19, 30], already available in educational or experimental tools[22].

We are also interested in involving the less formal practice of Design Thinking, as its bridging to CM has been at the core of the OMILAB Digital Innovation environment, specifically in the Scene2Model toolkit [43]. Design Thinking is a major preoccupation in our local industry, typically framed as a marketing practice rather than a CM preliminary effort of knowledge acquisition and ideation.

Threats. The new course design moves away from the joint ACM/AIS curricular recommendation in [2], as we assess the 2020 draft as requiring major revisions in the post-ChatGPT era and for a Business Administration school (i.e. going beyond software development towards enterprise systems viewpoints) and in the context of redefining the role of CM – "from representation to mediation" [7]

It is not clear how the industry and labor market assimilate the new skill profile emerging from this course re-design. It is too early for longitudinal assessments, but early signals show increasing interest on the companies' side in conceptualization and abstraction skills - especially as entry-level coding jobs start being delegated to GenAI products. Human-driven conceptualization is even in our outsourcing-dominated industry heavily involved in Product Development (from Design Thinking to Process Mapping), Robotic Process Automation (in the context of As-Is vs. To-Be business process analysis), Low-code Development (in the context of low-code platforms use and customization) and we hope for future longitudinal surveys to confirm the value of our multi-perspective CM offer.

As recommendations for potential adopters of similar strategies we formulate the following guidelines: introduce diverse modeling standards that show both semantic overlaps and distinctions; clarify their competency/scope boundaries regardless of visual customizations allowed by some tools; show the official specifications to reveal the importance of machine-readable serializations and the XML schemas governing them; bring forth a notion of model queries reflecting that of competency questions; pick educational tooling that easily demonstrate them through querying features or some model-based reporting (XPath may serve as a fallback demonstrator); employ standards that have reasonable visibility - start from UML, BPMN, Archimate, and expand to more specialized languages or DSMLs as time allows, maintaining focus on how this reflects in expanding competency and querying possibilities.

[21]https://www.omg.org/ekg/

[22]Bee-Up has been a long term promoter of the RDF export for a numbering of modeling languages and model types, thus allowing the model query to be demonstrated via SPARQL, see https://bee-up.omilab.org/activities/bee-up/

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] R. A. Buchmann, A.-M. Ghiran, Teaching knowledge graphs: A journey from logic to web development and semantics-driven engineering, in: Companion Proceedings of the 42nd International Conference on Conceptual Modeling: ER Forum, 7th SCME, volume 3618, CEUR-WS, 2023. URL: https://ceur-ws.org/Vol-3618/scme_paper_4.pdf.

[2] ACM/AIS IS 2020 Task Force, IS 2020 a competency model for undergraduate programs in information systems, 2020. URL: https://www.acm.org/binaries/content/assets/education/curricula-recommendations/is2020.pdf.

[3] J. Cabot, A. Vallecillo, Modeling should be an independent scientific discipline, Software and Systems Modeling 21 (2022) 2101—-2107. doi:10.1007/s10270-022-01035-8.

[4] I. Maslov, S. Poelmans, K. Rosenthal, Conceptual modeling education, Business & Information Systems Engineering (2025). doi:10.1007/s12599-025-00930-w.

[5] S. Strecker, U. Baumol, D. Karagiannis, A. Koschmider, M. Snoeck, R. Zarnekow, Five inspiring course (re-)designs, Business & Information Systems Engineering 61 (2019) 241–252. doi:10.1007/s12599-019-00584-5.

[6] D. Bork, A framework for teaching conceptual modeling and metamodeling based on bloom's revised taxonomy of educational objectives, in: Proceedings of HICSS 2019, AIS eLibrary, 2019. URL: https://aisel.aisnet.org/hicss-52/set/methods_models/7/.

[7] J. Recker, R. Lukyanenko, M. Jabbari, B. M. Samuel, A. Castellanos, From representation to mediation: a new agenda for conceptual modeling research in a digital world, Management Information Systems Quarterly 54 (2021) 269–300. doi:10.25300/MISQ/2021/16027.

[8] J. Michael, L. Cleophas, S. Zschaler, T. Clark, B. Combemale, T. Godfrey, D. Khelladi, V. Kulkarni, D. Lehner, B. Rumpe, M. Wimmer, A. Wortmann, S. Ali, B. Barn, I. Barosan, N. Bencomo, F. Bordeleau, G. Grossmann, G. Karsai, O. Kopp, B. Mitschang, P. Muñoz Ariza, A. Pierantonio, F. Polack, M. Riebisch, H. Schlingloff, M. Stumptner, A. Vallecillo, M. van den Brand, H. Vangheluwe, Model-driven engineering for digital twins: Opportunities and challenges, Systems Engineering 28 (2025) 659–670. doi:10.1002/sys.21815.

[9] R. A. Buchmann, A.-M. Ghiran, V. Döller, D. Karagiannis, Conceptual modeling education as a "design problem", Complex Systems Informatics and Modeling Quarterly 21 (2019) 21–33. doi:10.7250/csimq.2019-21.02.

[10] R. J. Wieringa, Design Science Methodology for Information Systems and Software Engineering, Springer, 2014. doi:10.1007/978-3-662-43839-8.

[11] R. A. Buchmann, The purpose-specificity framework for domain-specific conceptual modeling, in: D. Karagiannis, M. Lee, K. Hinkelmann, W. Utz (Eds.), Domain-Specific Conceptual Modeling, Springer, 2022, pp. 67–92. doi:10.1007/978-3-030-93547-4_4.

[12] D. Karagiannis, R. A. Buchmann, W. Utz, The OMiLAB digital innovation environment: Agile conceptual models to bridge business value with digital and physical twins for product-service systems development, Comput. Ind. 138 (2022). doi:10.1016/j.compind.2022.103631.

[13] M. Ullrich, C. Houy, T. Stottrop, M. Striewe, B. Willems, P. Fettke, P. Loos, A. Oberweis, Automated assessment of conceptual models in education: A systematic literature review, Enterprise Modelling and Information Systems Architectures 18 (2023). doi:10.18417/emisa.18.2.

[14] D. R. Krathwohl, A revision of bloom's taxonomy: An overview, Theory Into Practice 41 (2002) 212–218. doi:10.1207/s15430421tip4104_2.

[15] J. Mylopoulos, Conceptual modeling and Telos1, in: P. Loucopoulos, R. Zicari (Eds.), Conceptual Modeling, Databases, and Case An integrated view of information systems development, New York: Wiley, 1992, p. 49–68. URL: https://www.cs.toronto.edu/~jm/2507S/Readings/CM+Telos.pdf.

[16] C. Soyka, N. Schaper, E. Bender, M. Striewe, M. Ullrich, Toward a competence model for graphical modeling, ACM Trans. Comput. Educ. 23 (2022). doi:10.1145/3567598.
[17] A. Polyvyanyy, Process querying: Methods, techniques, and applications, in: Process Querying Methods, Springer, 2021, pp. 511–524. doi:10.1007/978-3-030-92875-9_18.
[18] C. M. Keet, Z. C. Khan, Discerning and characterising types of competency questions for ontologies (2024). arXiv:2412.13688.
[19] S. Bachhofner, E. Kiesling, K. Revoredo, P. Waibel, A. Polleres, Automated process knowledge graph construction from bpmn models, in: Proceedings of DEXA 2022, Springer, 2022, p. 32–47. doi:10.1007/978-3-031-12423-5_3.
[20] M. Vidgof, S. Bachhofner, J. Mendling, Large language models for business process management: Opportunities and challenges, in: Proceedings of BPM 2023 Forum, Springer, 2023, pp. 107–123. doi:10.1007/978-3-031-41623-1_7.
[21] M. Dumas, F. Fournier, L. Limonad, A. Marrella, M. Montali, J. R. Rehse, R. Accorsi, D. Calvanese, G. D. Giacomo, D. Fahland, A. Gal, M. L. Rosa, H. Völzer, I. Weber, AI-augmented Business Process Management Systems: A Research Manifesto, ACM Transactions on Management Information Systems 14 (2023). doi:10.1145/3576047.
[22] A. Gutschmidt, B. Nast, Assessing model quality using large language models, in: Proceedings of PoEM 2024, volume 538 of *LNBIP*, Springer, 2025, pp. 105–122. doi:10.1007/978-3-031-77908-4_7.
[23] G. Challa, A. Gartini, F. Muff, H.-G. Fill, An architecture for integrating large language models into metamodeling platforms: The example of MM-AR, in: Proceedings of ER Forum 2024, volume 3849, CEUR-WS, 2024. URL: https://ceur-ws.org/Vol-3849/poster-demo1.pdf.
[24] F. Härer, Conceptual model interpreter for large language models, in: Proceedings of ER Forum 2023, volume 3618, CEUR-WS, 2023. URL: https://ceur-ws.org/Vol-3618/forum_paper_11.pdf.
[25] D. N. Dolha, R. A. Buchmann, Generative AI for BPMN process analysis: Experiments with multi-modal process representations, in: Proceedings of BIR 2024, volume 529 of *LNBIP*, Springer, 2024, pp. 19–35. doi:10.1007/978-3-031-71333-0_2.
[26] P. Delfmann, D. M. Riehle, S. Höhenberger, C. Corea, C. Drodt, The diagramed model query language 2.0: Design, implementation, and evaluation, in: A. Polyvyanyy (Ed.), Process Querying Methods, Springer, 2022, pp. 115–148. doi:10.1007/978-3-030-92875-9_5.
[27] D. Karagiannis, Agile modeling method engineering, in: Proceedings of PCI 2015, Association for Computing Machinery, 2015, p. 5–10. doi:10.1145/2801948.2802040.
[28] R. A. Buchmann, A.-M. Ghiran, Engineering the cooking recipe modelling method: a teaching experience report, in: CEUR-WS, volume 1999, 2017. URL: https://ceur-ws.org/Vol-1999/paper5.pdf.
[29] A.-M. Ghiran, C. C. Osman, R. A. Buchmann, Advancing conceptual modeling education towards a generalized model value proposition, in: Advances in Information Systems Development, volume 39 of *LNISO*, Springer, 2020, pp. 1–18. doi:10.1007/978-3-030-49644-9_1.
[30] D. Karagiannis, R. A. Buchmann, A proposal for deploying hybrid knowledge bases: the ADOxx-to-GraphDB interoperability case, in: Proceedings of HICSS 2018, AIS eLibrary, 2018. URL: https://aisel.aisnet.org/hicss-51/ks/ks_creation/4/.
[31] M. Winter, R. Pryss, M. Fink, M. Reichert, Towards measuring and quantifying the comprehensibility of process models: the process model comprehension framework, Inf Syst E-Bus Manage 21 (2023) 723–751. doi:10.1007/s10257-023-00642-2.
[32] D. Moody, J. van Hillegersberg, Evaluating the visual syntax of UML: An analysis of the cognitive effectiveness of the UML family of diagrams, in: Proceedings of SLE 2008, volume 5452 of *LNCS*, Springer, 2009, pp. 16–34. doi:10.1007/978-3-642-00434-6_3.
[33] K. Rosenthal, J. Wagner, S. Strecker, Modeling styles in conceptual data modeling: reflecting observations in a series of multimodel studies, in: Proceedings of ECIS 2022, AIS eLibrary, 2022. URL: https://aisel.aisnet.org/ecis2022_rp/73/.
[34] B. Ternes, S. Strecker, K. Rosenthal, H. Barth, A browser-based modeling tool for studying the learning of conceptual modeling based on a multi-modal data collection approach, in: Proceedings

of the 14th Internationale Tagung Wirtschaftsinformatik 2019, 2019, pp. 1998–2002. URL: https://aisel.aisnet.org/wi2019/specialtrack03/papers/7.

[35] D. Bogdanova, M. Snoeck, Domain modelling in bloom: Deciphering how we teach it, in: Proceedings of PoEM 2017, volume 305 of *LNBIP*, Springer, 2017, pp. 3–17. doi:10.1007/978-3-319-70241-4_1.

[36] K. Rosenthal, B. Ternes, S. Strecker, Learning conceptual modeling: structuring overview, research themes and paths for future research, in: Proceedings of ECIS 2019, Association for Information Systems, 2019. URL: https://aisel.aisnet.org/ecis2019_rp/137, paper 137.

[37] I. Maslov, S. Poelmans, Advancing the BPMN 2.0 standard with an extended animated notation: A research program for token-based process modeling education, in: Proceedings of BIR 2023 Workshops and Doctoral Consortium, volume 3514, 2023. URL: https://ceur-ws.org/Vol-3514/paper92.pdf.

[38] V. Richardson, Constructivist teaching and teacher education: Theory and practice, in: Constructivist Teacher Education: Building a World of New Understandings, Routledge, 1997, pp. 3–14.

[39] U. Frank, Conceptual modelling as the core of the information systems discipline–perspectives and epistemological challenges, in: Proceedings of AMCIS 1999, 1999, pp. 695–697. URL: https://aisel.aisnet.org/amcis1999/240.

[40] J. S. Bruner, The Process of Education, Harvard University Press, 1960.

[41] Ş. Uifălean, R. A. Buchmann, Low-code browser front-end automation using RDF graphs and a domain-specific language for UX representation, in: Research Challenges in Information Science (RCIS) 2025, volume 547 of *LNBIP*, Springer, 2025, pp. 140–155. doi:10.1007/978-3-031-92474-3_9.

[42] G. Guizzardi, G. Wagner, R. Guizzardi, Towards ontological foundation for conceptual modeling: The unified foundational ontology (UFO) story, Applied Ontology 10 (2015) 259–271. doi:10.3233/AO-150157.

[43] C. Muck, S. Palkovits-Rauter, Conceptualizing design thinking artefacts: The scene2model storyboard approach, in: Domain-Specific Conceptual Modeling: Concepts, Methods and ADOxx Tools, Springer, 2022, pp. 567–587. doi:10.1007/978-3-030-93547-4_25.