Information and Software Technology 183 (2025) 107736

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Unraveling the pain points of domain modeling

Isadora Valle [a,*], Tiago Prince Sales [a], Eduardo Guerra [c], Maya Daneva [a], Renata Guizzardi [b], Luiz Olavo Bonino da Silva Santos [a], Henderik A. Proper [d], Giancarlo Guizzardi [a]

[a] Semantics, Cybersecurity & Services (SCS), University of Twente, Hallenweg 19, Enschede, 7522 NH, The Netherlands
[b] Industrial Engineering and Business Information Systems (IEBIS), University of Twente, Hallenweg 19, Enschede, 7522 NH, The Netherlands
[c] Faculty of Engineering, Free University of Bozen-Bolzano, Piazza Domenicani, 3, Bolzano, 39100 BZ, Italy
[d] Business Informatics Group (BIG), TU Wien, Favoritenstraße 9-11, Vienna, 1040, Austria

ARTICLE INFO

Keywords:
Conceptual modeling
Pain points
Modeling costs
Customer journey map
Multi-method research approach

ABSTRACT

Conceptual models offer numerous benefits but require significant investments, requiring modelers to strive to balance costs and benefits. Understanding the modeling process and the frustrations experienced by modelers can provide valuable insights for this assessment. While research acknowledges certain instances of modelers' dissatisfaction, its scope often limits detailed examination. This study seeks to identify and analyze the main pain points associated with domain modeling through a five-phase empirical study using a multi-method approach. We identified **71** pain points, synthesized them to **41**, and prioritized **16** as the most significant and prevalent in domain modeling. We then refined, documented, and exemplified the prioritized pain points, analyzed their potential causes, and discussed their practical implications. Our findings provide valuable insights for improving modelers' experiences and optimizing the modeling process.

1. Introduction

Conceptual models are abstractions that facilitate "understanding and communication of the physical and social world" [1]. They support problem-solving, decision-making, communication, information systems engineering, and other purposes [2]. Nevertheless, developing, managing, and implementing these models demands significant effort [3], requiring modelers to strive to balance costs and benefits. Despite their importance, in-depth research on the modeling process and the challenges modelers face remains limited [4]. Although several studies acknowledge these challenges (Section 8), they often overlook their causes, consequences, and potential solutions.

This article addresses these gaps by investigating the challenges faced by modelers involved in domain modeling. We refer to them as "pain points", a term from Marketing that describes customer dissatisfaction when expectations are unmet [5]. Empirical studies have applied pain point analysis in diverse areas of computer science. For instance, a taxonomy of 2,110 reported pain points in scientific software development was proposed in [6], nine challenges related to computational notebooks were identified in [7], 21 pain points in the context of requirements engineering were analyzed in [8], and 38 issues concerning agile-hybrid project management were documented in [9].

To examine pain points in domain modeling, we conceptualize modeling as a dynamic interaction between different products (e.g., models, languages, tools), activities (e.g., requirements elicitation, concept negotiation), and agents (e.g., modelers, domain experts, sponsors). From this perspective, modeling can be seen as a service process in which modelers can assume a dual role as service providers and consumers. This study focuses on identifying instances of dissatisfaction that arise when modelers, acting as service consumers, interact with other elements of the modeling process.

Our investigation was driven by the following research question: *What are the main pain points associated with domain modeling?*. Understanding these pain points can enhance the modeling process by supporting (i) planning initiatives, (ii) assessing costs, (iii) identifying solutions, (iv) improving methods, languages, and tools, and (v) guiding novice modelers. Prior research highlights the value of addressing frustrations to improve processes [10], and the importance of managing stakeholders' emotions in software development [11]. Similarly, we argue that addressing modelers' pain points can lead to more effective modeling outcomes.

* Corresponding author.
E-mail addresses: i.vallesousa@utwente.nl (I. Valle), t.princesales@utwente.nl (T.P. Sales), eduardo.guerra@unibz.it (E. Guerra), m.daneva@utwente.nl (M. Daneva), r.guizzardi@utwente.nl (R. Guizzardi), l.o.boninodasilvasantos@utwente.nl (L.O. Bonino da Silva Santos), henderik.proper@tuwien.ac.at (H.A. Proper), g.guizzardi@utwente.nl (G. Guizzardi).

https://doi.org/10.1016/j.infsof.2025.107736
Received 2 December 2024; Received in revised form 18 March 2025; Accepted 25 March 2025
Available online 9 April 2025

To answer this research question, we conducted a qualitative empirical study using a multi-method approach. Initially, we designed an expected journey of a domain modeler to guide the tasks undertaken in the subsequent phases. Secondly, we performed a thematic analysis [12], a virtual brainstorming session [13], and a purposive literature review [14], identifying 71 pain points. Then, we employed qualitative data synthesis to condense these pain points to 41. Subsequently, we conducted a two-round Delphi study [15] to prioritize them, identifying 16 as the most prevalent and impactful in the modeling experience. Finally, we refined, documented, and exemplified each prioritized pain point, identifying underlying causes and discussing practical implications. We focused on uncovering the causes of the pain points rather than proposing solutions because effective responses are highly context-dependent, varying with model type, project constraints, and organizational environments. This emphasis on causes offers practitioners insights to craft solutions suited to their contexts.

The remainder of the paper unfolds as follows. Section 2 delves deeper into the definitions of models, modeling, and customer journeys. Section 3 describes the multi-method research process and materials employed within the study. Section 4 presents and discusses the pain points, while Section 5 analyzes their possible causes. Section 6 reflects on the implications of our results for research and practice. Section 7 is about validity threats. Section 8 contextualizes our work within the current state of the art. Finally, Section 9 concludes and discusses the direction of future work.

2. Background

2.1. On models and modeling

A domain model is "a *social artifact* that is *acknowledged* by a *collective agent* to *represent* an *abstraction* of some *domain* for a particular *cognitive purpose*" [16]. Models are *social artifacts* in the sense that their role should be identifiable by multiple individuals. Although the concept of a *collective agent* implies that the creation and use of domain models typically involve multiple people, the special case of a "self-shared" model, where a domain model is utilized by an individual alone, is also recognized. *Abstraction* implies that, in alignment with the model's cognitive purpose, modelers must determine which aspects of the domain to represent. In this definition, *domain* refers to any subject of discussion or reflection. It encompasses anything modeled, whether it exists in the real world, is intended to exist in the future, or is imaginary. *Cognitive purpose* implies that a model must be created with a specific cognitive goal, such as expressing, specifying, or experiencing knowledge about the domain being modeled. Consequently, domain models can range from simple drawings, such as a house sketch on paper or a workflow diagram in PowerPoint, to traditional models used in model-based development, such as a class diagram or a statechart, and further to more complex representations, including process models and blueprints. We even consider within this scope non-traditional models such as those represented in domain-specific languages as well as metamodels defining the abstract syntax of such languages.

A conceptual model is a type of domain model. They are *information objects* intentionally created to describe the *mental models* of a given domain [17]. As an *information object*, this model results from an intentional act, meaning it is created to describe a conceptualized reality through individual and relational concepts. These representational primitives, called concepts, act as the "lens" through which reality is perceived and organized, reflecting regularities in reality that are cognitively relevant to us. For instance, consider navigating a city using the subway system. By employing concepts such as subway line, subway station, line direction, order of stations, and intersections between lines, we create a conceptual mental representation of the subway network. These mental representations can be thought of as *mental models*, which are personal and partial accounts of external reality, filtered through the lens of conceptualization, that individuals use to interact with the world around them. Thus, conceptual models reflect a conceptualization, which gives them conceptual semantics, and to which they make an inevitable ontological commitment. They have an advantage over natural language or diagrammatic notations because they are based on formal notation that captures the semantics of the application. Moreover, as descriptions of mental models, conceptual models are not limited to graphical representations, nor are they restricted to modeling structural aspects of a domain or representing types. Examples of conceptual models include goal models (e.g., in i* [18]), business process models (e.g., in BPMN [19]), value models (e.g., in e3value [20]), models focusing on instances (e.g., city maps, subway maps), and models mixing instances and types (e.g., enterprise models in ArchiMate [21]).

Structural conceptual models are a subset of conceptual models that aim to identify, analyze, and describe *key structural regularities* within a specific universe of discourse [22]. These regularities include types, attributes, relations, and constraints. Examples of such models include those specified using Entity Relationship (ER; [23]), UML Class Diagrams [24], OntoUML [22], and Object-Role Modeling (ORM; [25]). These models are the focus of our analysis.

In software development, structural conceptual models are applied within two distinct contexts: Model-Based Development (MBD) and Model-Driven Development (MDD) [26]. In MBD, models facilitate understanding, documentation, and communication, with manual translation into code. Conversely, in MDD (or low-code environments), models serve as the foundation for the entire development lifecycle, including (semi-)automated code generation. Given that these contexts entail different model usages, which substantially influence the pain points encountered by modelers, our analysis focus on the MBD approach.

Henceforth, any references to models, modelers, or modeling should be interpreted within the framework of structural conceptual modeling for model-based development. For simplicity, the term "modelers" is used to refer to individual practitioners and modeling teams.

2.2. Customer journeys

To interpret and analyze pain points, it is essential to contextualize them. Considering modelers' activities and goals when emotions arise facilitates understanding their frustration. Therefore, to support our analysis, we develop the Customer Journey Map (CJM) of a modeler using the Customer Journey Modeling Language (CJML) [27].

Customer journey maps are a marketing tool used by organizations to analyze interactions from the customer's perspective [28]. These maps outline customers' actions, needs, and perceptions across their engagement with a company, capturing all potential touchpoints throughout the service experience. Their widespread adoption stems from their ability to clarify these interactions, enabling organizations to implement solutions that enhance the customer experience.

Customers approach each interaction with specific expectations, and when these expectations are unmet, pain points emerge from their experience. These pains are emotional rather than physical, resulting from customers' psychological dissatisfaction when their expectations are not met while interacting with a service or product [5]. Identifying and mapping these instances of dissatisfaction is challenging due to the variability of individual experiences and users' tendency not to express concerns through explicit complaints or negative feedback. Therefore, when analyzing customer touchpoints to uncover pain points, several questions must be considered: Who are the individuals engaging in these interactions? What is significant for each persona? What were their experiences before this journey? What information have they been exposed to, and what do they already know? It is imperative to empathize with the user's perspective to grasp how each interaction conflicts with their expectations.

In Fig. 1, we present an example of a customer journey map developed with the CJML [27]. To interpret this map, it is essential to understand three key concepts in the CJML: *actor*, *touchpoints*, and

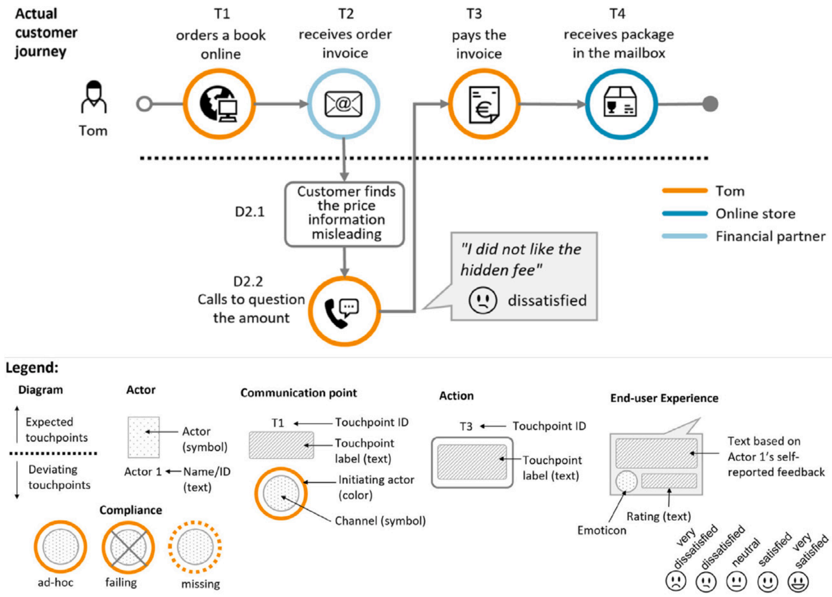

Fig. 1. Customer journey map from a bookstore adapted from [27].

end-user experience. An *actor* is an icon identifying any individual or entity, such as an organization or company, involved in delivering a service. *Touchpoints*, often understood as stages in a customer journey, are categorized into two types: communication points (circles) and actions (squircle). Communication points represent instances of interaction between two actors, while actions denote events or activities undertaken by the customer or service provider. The *end-user experience* is a textbox wherein the user's experience is articulated, encompassing textual descriptions, ratings, and emoticons reflecting their emotions. Within this context, we characterized pain points as instances where the user's experience is rated as "very dissatisfied" or "dissatisfied".

The map in Fig. 1 illustrates Tom's journey while purchasing a book online. In the figure, touchpoints above the dotted line are expected by the company, while those below represent unexpected events. Therefore, Tom's journey comprises six touchpoints: T1, T2, T3, T4, D2.1, and D2.2, with D2.1 and D2.2 representing deviations from the expected journey outlined by the company. This deviation occurred because Tom found the price information on the order invoice misleading. After contacting the company to address his concerns, Tom expressed frustration over unclear fees. This instance of dissatisfaction experienced by Tom constitutes what we call a pain point.

Pain point analysis should identify three components: cause, description, and consequences. In the example given, the cause is the discrepancy between the invoice and purchase order details, leading to unclear fees. The pain point reflects Tom's emotional response, including frustration and a sense of being deceived. Consequences may include Tom canceling the purchase or pursuing legal action.

Pain points often arise when deviations occur from the anticipated journey map, but they can also emerge even when customers follow the expected path. This is because, although companies design journey maps to anticipate and address potential issues, not all sources of customer dissatisfaction can be fully mitigated.

Customer experience analysis in conceptual modeling can be approached by viewing the modeling process as a dynamic interaction among products, activities, and agents. Products include outcomes like the model and documentation, as well as tools, languages, techniques, and methods used in its development. Activities refer to tasks such as model creation, implementation, and maintenance. Agents are individuals involved with these products and services, including modelers, sponsors, users, collaborators, and suppliers.

By conceptualizing the modeling process in this manner and mapping its elements, it became evident that it could be treated as a distinct service process. This process is characterized by a succession of instantaneous activities, inherently perishable, executed with specialized expertise, wherein interactions among agents play a pivotal role in shaping the process outcome [29]. Within this notion, modelers can assume a dual role as service providers and consumers. As providers, modelers deliver a modeling service that produces a model as a product. As consumers, modelers engage with methods, tools, languages, and other services provided by various agents. In this study, *we focused exclusively on modelers' experience from the perspective of service consumers.*

3. Research process and materials

Our study is exploratory and adopts a qualitative multi-method research approach [30]. We set up a research process structured in seven stages, grouped into four phases and five methods, aimed at discovering, synthesizing, and prioritizing the pain points of modeling, as outlined in Fig. 2. Study findings and replication materials are available in the replication package [31].

In the first phase, a journey map of a modeler was created to guide the identification, analysis, and exemplification of pain points. Although refined throughout the study, this map was not formally validated and is used only as a supporting material to enhance the understanding of the pain points. The second phase identified pain points in modeling by employing thematic analysis [12], virtual brainstorming sessions [13], and a purposive literature review [32]. The third phase synthesized these pain points using qualitative data synthesis [33]. In the fourth phase, they were prioritized based on their frequency and impact through a Delphi study [15] involving a voting and a prioritization session. Finally, the fifth phase refined, documented, and exemplified the prioritized pain points.

3.1. Phase 1: Designing the expected journey of a modeler

We designed the expected journey map of a modeler, as depicted in Fig. 3, to guide our pain point identification, analysis, and exemplification. This map was crafted based on our own experiences as modelers, as well as insights gleaned from established methodologies such as the Systematic Approach for Building Ontologies (SABiO; [34]), the NeOn Methodology [35], the Nijssen Information Analysis Method (NIAM; [36]), the Conceptual Schema Desing Procedure (CSDP; [25]),

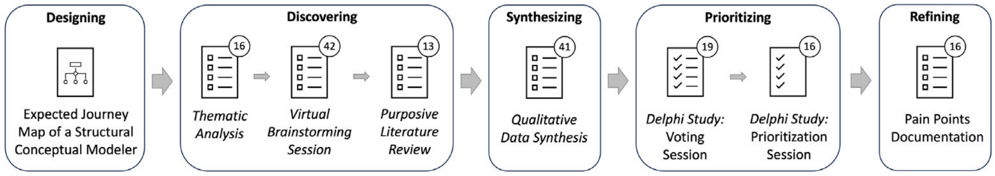

Fig. 2. Research methodology process and findings.

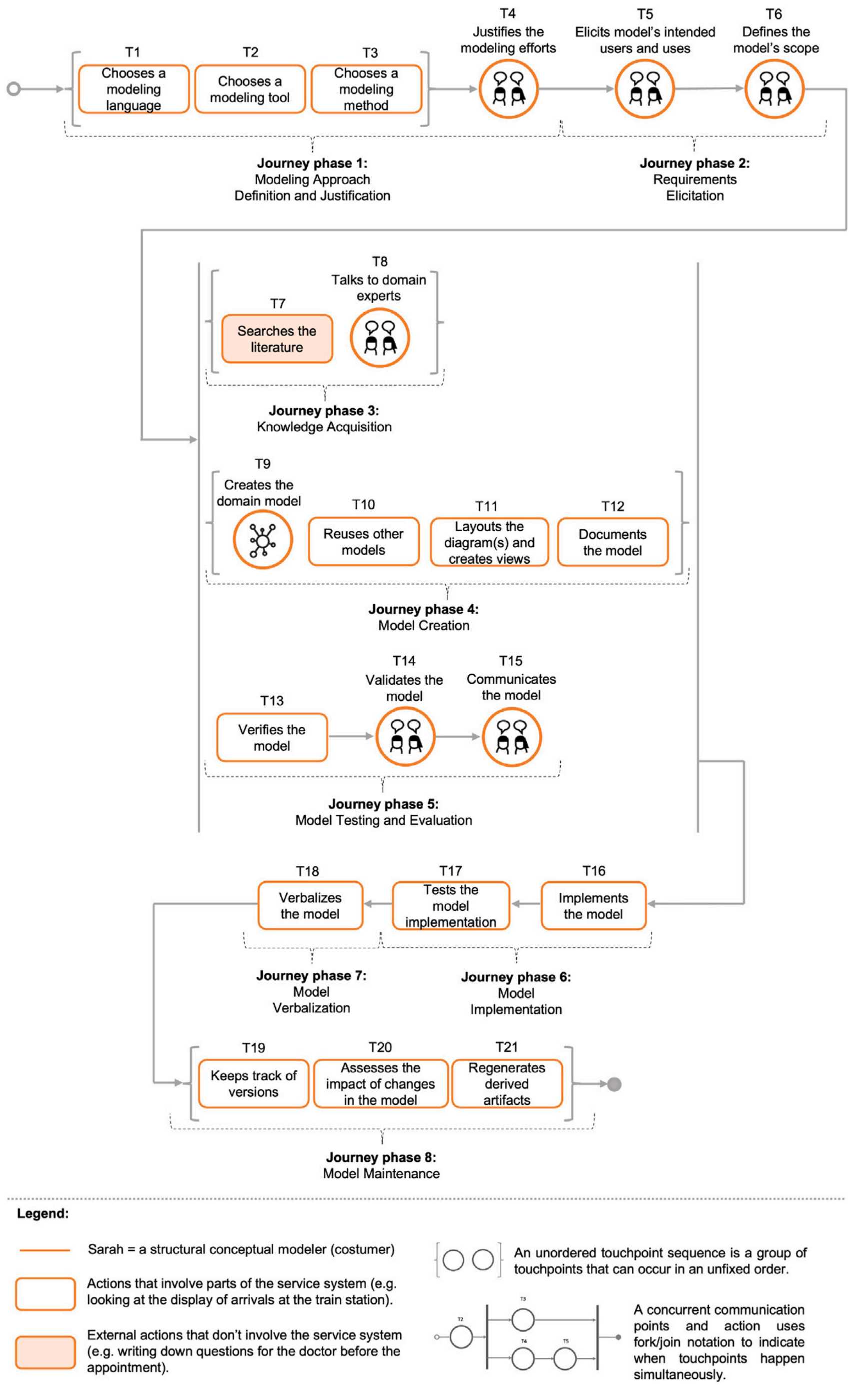

Fig. 3. Expected journey map of a modeler.

and the Rational Unified Process [37]. The outlined journey was developed using the CJML [27] and consists of eight distinct phases encompassing 21 touchpoints, including 8 communication points, 12 actions, and 1 external action.

The journey depicted in Fig. 3 is linear to simplify the analysis of pain points, though real-world modeling processes are typically more cyclical and concurrent. While we used notation to indicate concurrent actions in Phases 3, 4, and 5, this linear representation does not fully capture the dynamic nature of modeling. As such, when analyzing this journey, it is important to consider that the phases may overlap, repeat, or occur in a different sequence depending on the context. Although the modeling process is not standardized, modelers' goals generally remain consistent across different approaches. Therefore, to enhance understanding of this journey, we defined the goals and motivations of modelers for each phase.

Phase 1: Modeling Approach Definition and Justification Modelers aim to select the appropriate modeling language, tools, and methods, demonstrating the suitability of the model-based approach for addressing the problem.

Phase 2: Requirements Elicitation Modelers aim to define the model's purpose and intended applications, translating them into requirements that delineate its scope.

Phase 3: Knowledge Acquisition Modelers seek to gather domain-specific knowledge, drawing insights from experts and consolidated bibliographic resources.

Phase 4: Model Creation Modelers aim to construct the model by identifying and organizing key domain elements, creating model views, and documenting definitions. They also reuse existing conceptualizations when possible.

Phase 5: Model Testing and Evaluation Modelers seek to verify and validate the model, ensuring it is accurately constructed and suitable for the intended purpose, through automated tools or expert feedback.

Phase 6: Model Implementation Modelers aim to convert the model's conceptual specifications into practical design specifications, which are then tested and implemented in specific application contexts.

Phase 7: Model Verbalization Modelers aim to translate the model into a format suitable for the target audience to improve stakeholders' comprehension.

Phase 8: Model Maintenance Modelers seek to apply strategies for managing modeling outputs, focusing on handling updates, changes, and version control.

3.2. Phase 2: Discovering pain points

3.2.1. Thematic analysis

This study employed thematic analysis [12] to explore pain points experienced by modelers. The analysis drew on anonymized transcripts of online interviews conducted by the first author between November 2022 and January 2023 as part of a prior study presented in [2].

These transcripts followed the interview content, which covered participants' backgrounds, specific modeling projects, modeling processes, and broader perspectives on modeling, such as costs, benefits, and return on modeling effort. They were available in English, Portuguese, or Italian and analyzed in the original language. They included accounts from nine experienced modelers selected randomly from relevant literature authors, professional networks, and personal contacts. These participants had substantial expertise, including advanced degrees in computer science-related fields, proficiency in multiple modeling languages, and involvement in over 20 modeling projects across diverse domains. They represented diverse roles within organizational settings, including universities (4), large (3) and small (2) companies, located in Brazil (4), the Netherlands (4), and Italy (1).

The thematic analysis was conducted using Nvivo 12, following the process outlined in [12]. The first author initially familiarized herself with the data through comprehensive reading, followed by deductive coding to identify and categorize meaningful data segments. Preliminary codes were refined and validated collaboratively within the research team. Themes were then developed, reviewed for coherence, and mapped. The second author independently validated the themes and coded segments, with a third author mediating disagreements. This process identified 16 pain points [31].

As the analysis was based on pre-existing data, saturation could not guide sample size determination [38]. Additionally, the broad scope of the interviews raised questions about the representativeness of the findings. To mitigate these limitations, we conducted a virtual brainstorming session with modelers and performed a purposive literature review, providing additional insights to refine the analysis.

3.2.2. Virtual brainstorming session

The first author conducted a virtual brainstorming session following the methodology outlined in [13]. This approach encouraged idea generation by prioritizing quantity, fostering a non-judgmental environment, and supporting creative exploration.

The session included six participants selected to ensure a balanced representation across two dimensions: domain modeling experience (beginners[1] and experts[2]) and professional background (practitioners and academics). None had been involved in the earlier interviews. The session began with an introductory meeting to explain the study's purpose, define pain points, and address procedural queries. Due to the participants' geographic dispersion (Brazil and the Netherlands), the meeting was held in a hybrid format using Microsoft Teams. The brainstorming session itself was conducted asynchronously on the Mural platform, where participants reviewed previously identified pain points and added new ones using color-coded Post-it notes. Each participant was assigned a unique color for identification purposes. To maintain anonymity, the color coding of participants' Post-it notes was managed by the first author. To aid participants unfamiliar with pain point analysis, the virtual workspace was structured to include descriptions of the modeling phases and their goals [31], inspired by the customer journey map presented in Phase 1. This helped participants reflect on their modeling experiences and identify challenges.

Although the activity appeared individualistic, it retained an interactive element through platform features allowing participants to view and comment on each other's contributions. Over a week-long period, this collaborative environment facilitated knowledge sharing and enriched the session's outcomes. Following the session, the first author reviewed all contributions, conducted follow-up discussions with participants for clarification, and ensured alignment with the established pain point format. The virtual brainstorming session resulted in 42 new pain points [31].

3.2.3. Purposive literature review

A purposive literature review [32] was conducted to uncover additional pain points overlooked by earlier methods. The subjective and underexplored nature of the term "pain points" posed challenges for retrieving relevant data using conventional search engines. To navigate this issue, the review prioritized a broad and flexible approach, diverging from the exhaustive coverage typical of systematic reviews. This strategy enabled the inclusion of insights from familiar sources and papers identified through snowballing techniques [39], including backward and forward citation tracking, until saturation was achieved [38].

Given the limited direct references to "pain points" in the conceptual modeling literature, two questions were addressed before analyzing the papers: (A) *How can a pain point be identified in a text fragment that lacks a personal narrative, given its connection to emotional responses?*, and (B) *What alternative terminologies might convey the notion of "pain points", given its infrequent use in the domain?*.

To address the first question, we applied a lexicon-based emotion detection technique [40]. Based on the WordNet-Affect taxonomy [41], we identified "frustration" as the emotion most closely associated with a pain point. Using Thesaurus.com [42], we selected the strongest matches synonyms for this emotion. The resulting keywords — *annoyance, dissatisfaction, failure, grievance, irritation, and resentment* — formed the set used to identify emotional responses to pain points in textual contexts.

To address the second query, we also employed a lexicon-based approach, beginning with Dictionary.com's definition of "pain point"

[1] Beginner: Basic understanding, may require guidance, developing expertise, capable of simple models.

[2] Expert: Highly proficient and sought after, handles complex tasks, in-depth understanding of methodologies and tools.

as a problem, frustration, or troublesome issue [43]. From this definition, we determined that lexical equivalents of “problem” effectively conveyed the concept of a pain point.[3] Using Thesaurus.com [42], we selected the strongest matches synonyms for this term, removing redundancies. The resulting keywords — *complication, dilemma, dispute, headache, issue, obstacle, question, and trouble* — formed the set used to identify pain points in textual contexts.

Building on the defined keywords, the first author manually reviewed the curated papers, identifying occurrences of these terms and analyzing their context to extract pain points. To ensure reliability, a second author validated the findings. This process identified 13 additional pain points [31] and reinforced findings from earlier phases.

3.3. Phase 3: Synthesizing pain points

In reviewing the pain points derived from Phase 2, we identified a need for refinement. This issue is common in qualitative studies, as isolated studies may not contribute significantly to the understanding of a phenomenon. Synthesizing findings from multiple studies improves the robustness and transferability of results [33]. Therefore, we applied a qualitative data synthesis method, specifically a meta-synthesis approach, to integrate the identified pain points.

The synthesis process followed a four-phase framework proposed in [33]. First, we compiled the 71 pain points within a virtual workspace on the Mural platform [31]. The first two authors then read and analyzed the data, distinguishing key findings from examples representing potential causes and consequences. Next, they reorganized the data by grouping similar results and associating each example with one or more pain points. Finally, analogous pain points and their corresponding examples were merged to form unified descriptions. This process resulted in 41 synthesized pain points [31].

3.4. Phase 4: Prioritizing pain points

To prioritize the synthesized pain points, we applied the Delphi method [15]. This method enabled us to collect domain modelers’ opinions on the identified pain points, helping determine which were most frequent and which had the greatest impact on their modeling experiences. The process was interactive, ensuring participant anonymity, offering controlled feedback, and culminating in a final report.

For participant selection, we applied the same criteria used in the virtual brainstorming session, recruiting six modelers, including three experts and three beginners, with diverse backgrounds in academia and industry. These participants had experience working on projects in Brazil and Italy and were distinct from those involved in the brainstorming session and the interviews. Selection prioritized representativeness by focusing on participants’ subject-matter expertise [44] and contrasting the perspectives of experts with those of non-experts [45], rather than relying on statistically representative samples. The Delphi process, in which these participants were involved, was structured into two rounds: voting and prioritization. During the rounds, to ensure anonymity, participants’ names were coded, and results were visible only to the first author.

3.4.1. Delphi study: Voting sessions

For the voting session, a virtual workspace was created on the Mural platform [31], displaying the 41 pain points identified in the synthesis phase. Participants were asked to individually and asynchronously vote on pain points they had personally experienced during domain modeling. The first author supervised each session to assist with navigation and clarifications. Each session lasted less than 40 min, and the results are available in [31]. All votes were counted, and only the 19 pain points with at least two votes were selected for the next round.

[3] The term “Troublesome Issue” was omitted for lacking synonyms on Thesaurus.com, and “Troublesome” was excluded as it is an adjective, a linguistic category typically avoided in academic writing.

3.4.2. Delphi study: Prioritization sessions

For the prioritization sessions, a new virtual workspace was set up on the Mural platform [31], displaying the 19 pain points from the first round along with a Frequency X Impact matrix. The matrix included classifications for frequency — Always, Sometimes, Rarely — and impact — Low, Moderate, High. Participants were asked to individually and asynchronously categorize each pain point based on their responses to two questions: “Throughout your career, how often did you experience this frustration?” and “On average, how frustrated did you get?”. The first author supervised each session to assist with navigation and clarifications. Each session lasted less than 40 min. The results were then quantified using a weighted system and analyzed separately for experts and beginners. The prioritization outcomes and their quantification are available in [31].

As presented in Table 1, only the 16 pain points prioritized with at least one score equal to four and one higher than four per modeler group were included in our analysis. This selection criterion concentrated the analysis on pain points with consistently higher scores, as lower-scoring issues (e.g., scores of 2 or 3) may reflect less critical concerns. Furthermore, it allowed for the inclusion of pain points that may be perceived differently across groups. Upon examining the results from beginners (B) and experts (E), it becomes apparent that pain points PP01 through PP08 were prioritized by both groups. Meanwhile, pain points PP09, PP10, PP12, PP13, and PP15 were prioritized by beginners, whereas pain points PP11, PP14, and PP16 were concerns emphasized by experts.

3.5. Phase 5: Refining the prioritized pain points

In this phase, we refined our analysis by documenting the 16 pain points prioritized in Phase 4, drawing on our technical knowledge, practical experience, insights from the employed research methods, and relevant literature. The documentation includes detailed explanations, examples, and discussions on potential causes and consequences of each pain point.

Documenting the pain points involved the first three authors and four phases. First, they adopted the modeler’s perspective to understand their frustrations, addressing questions such as *“Why did they feel frustrated?”*, *“What did they feel frustrated about?”*, and *“What did they do after they felt frustrated?”*. Next, they analyzed the consequences of the pain points in the modeling process. They then consulted the literature to find evidence to support their assumptions and used their experience and case studies to identify illustrative situations for each pain point. Lastly, they used the CJML [27] to create visual representations of these situations. The final documentation is presented in Section 4.

4. Paint points of domain modeling

PP01 (Improper Requirements): “I could not properly define the model requirements.”

This pain point is associated with a modeler’s frustration in realizing that the model’s requirements were inadequately defined. Well-defined requirements exhibit qualities such as cohesiveness, completeness, consistency, correctness, relevance, usability, verifiability, and others [46]. Consider, for instance, a modeler developing a conceptual model to integrate two systems. During the requirements phase, the modeler defines classes and attributes based on data integration needs, data origins, and data relationships. However, issues emerge in the integration phase, revealing overlooked data and missing classes or attributes. As shown in Fig. 4, this scenario underscores deficiencies in the initial requirements definition, necessitating rework across multiple stages of the modeling process.

Failures in specifying model requirements are common, largely due to the involvement of individuals without adequate training in

Table 1
Results of the prioritization sessions.

Cod	Pain Point	Mean	P1_B1	P2_B2	P3_B3	P4_E1	P5_E2	P6_E3
PP01	I could not properly define the model requirements because, *e.g., my stakeholders did not know exactly what they needed or the domain was unfamiliar to me.*	5,2	6	5	6	4	6	4
PP02	My stakeholders resisted adopting modeling, a modeling language, or a tool because, *e.g., they did not think it was worth it or I could not show its value.*	4,8	4	6	6	2	5	6
PP03	It was costly to define the model requirements because, *e.g., there were too many stakeholders.*	4,8	5	4	6	6	6	2
PP04	It was costly to negotiate a common definition among the experts about a concept of the project's domain of interest	4,8	5	5	6	4	5	4
PP05	It was costly to generate verbalizations of my model suitable for the different audiences that should be able to read and use it.	4,7	5	4	5	6	5	3
PP06	I had difficulty explaining my model to my stakeholders/domain experts.	4,5	5	6	4	6	2	4
PP07	Writing definitions for the elements in my model was hard and boring.	4,3	4	4	6	5	3	4
PP08	The model has become too complex and too costly to maintain because, *e.g., the tool did not support documentation or there were too many elements.*	4,3	3	5	4	5	5	4
PP09	I wanted to reuse an existing model, but I had to redraw it from scratch.	4,3	3	4	3	5	6	5
PP10	I could not choose the modeling tool I wanted because of, *e.g., cost, expertise, culture, or bureaucracy.*	4,2	4	4	2	4	6	5
PP11	It was costly to manually split the model into visually appealing and understandable views (diagrams).	4,0	5	6	3	5	3	2
PP12	My colleagues did not consider it important to document the model and did not understand my efforts to do so.	4,0	5	3	3	3	4	6
PP13	The tool I chose did not support model verification. My model was large and complex and verifying it by myself was not trivial.	3,8	3	4	2	4	5	5
PP14	It was costly to define the model-driven approach because, *e.g., my company did not have a modeling culture or I was not knowledgeable enough.*.	3,5	4	5	2	2	4	4
PP15	I was the only one on the team interested in using and reusing the model.	3,5	5	2	2	3	5	4
PP16	I was not sure how to document my model.	3,3	4	3	5	4	2	2
PP17	I was the only one on the team with experience in modeling. So I did everything myself. I wish I had had someone to discuss the challenges.	3,3	4	3	2	3	5	3
PP18	I felt like I was repeating myself when manually creating my database schema (or OWL vocabulary).	3,3	4	4	2	3	2	5
PP19	I could not make one of my diagrams look good.	2,8	5	2	2	3	3	2

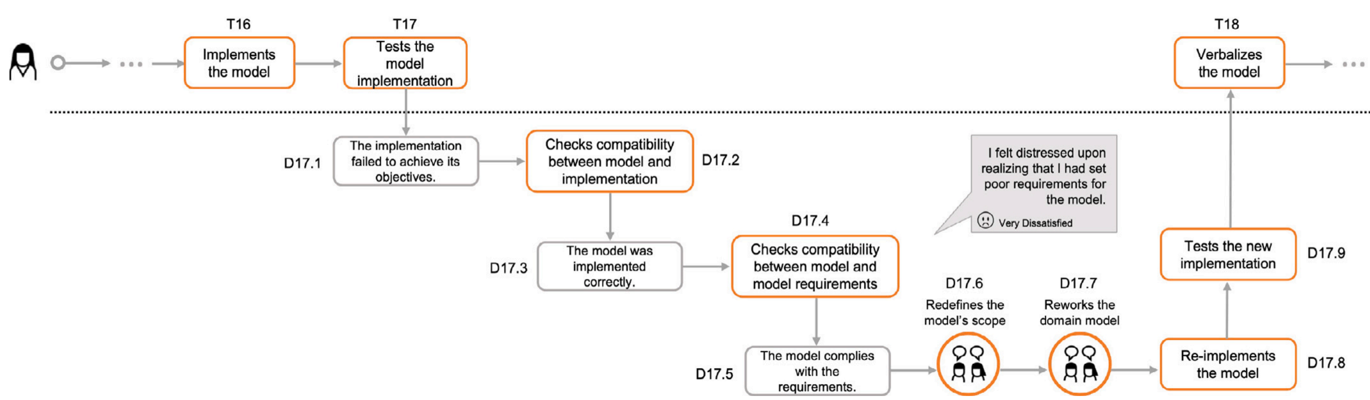

Fig. 4. Example of an occurrence of *P01 (Improper Requirements)*.

requirements engineering, such as managers, domain experts, and modelers. While the principles for specifying requirements are straightforward once learned, empirical evidence suggests that the process is non-intuitive for most [46]. Addressing this pain point can help modelers acquire the necessary skills to define well-structured requirements, reducing future frustrations. However, even experts in requirements definition may face challenges if they lack modeling skills, as model requirements involve considerations such as model type, domain, goals, and tools. For example, in ontology development, modelers use competency questions to define models' scope, yet empirical research shows that challenges remain in formulating and managing these questions [47]. Project-related or domain-specific challenges, such as unclear stakeholder needs or domain unfamiliarity, can also contribute to this pain point. Employing a proper requirements engineering method can mitigate these issues, but choosing an unfamiliar or inadequate method may lead to poor requirements definition.

Note that this pain point is inevitable in interactive approaches in which requirements must evolve with the model's development. In these cases, the modeling process itself can contribute significantly to shaping the requirements, helping to minimize emotional distress by promoting a more flexible and iterative approach.

PP02 (Resistance to Modeling): "My stakeholders resisted adopting modeling, a modeling language, or a tool."

This pain point is about the frustration of a modeler when facing resistance from other team members (e.g. managers, developers, domain experts) toward adopting a (structural conceptual) model-based solution or its components. Such resistance often arises among individuals with either previous negative experiences with modeling or a lack of familiarity with it. In the first instance, team members may have participated in projects where models were poorly implemented, offered minimal value, or were perceived as bureaucratic formalities. In the second, as illustrated in Fig. 5, they are experiencing neophobia, a psychological defense mechanism characterized by fear or aversion to novelty commonly observed in human behavior across diverse contexts [48]. Resistance also exists among individuals in certain fields. For example, in Agile software development, the value of using conceptual

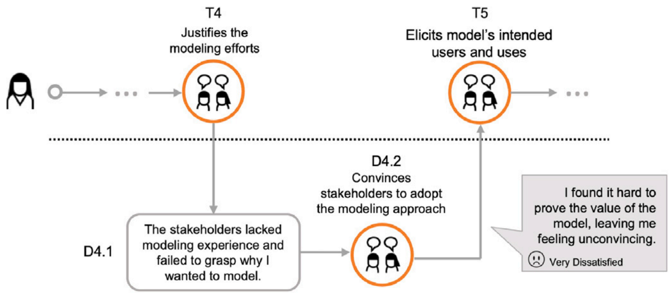

Fig. 5. Example of an occurrence of *PP02 (Resistence to Modeling)*.

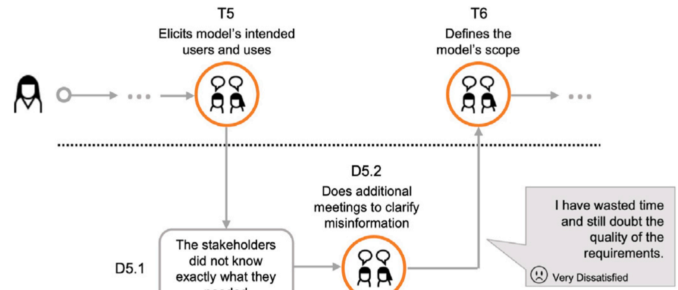

Fig. 6. Example of an occurrence of *PP03 (Expensive Requirements)*.

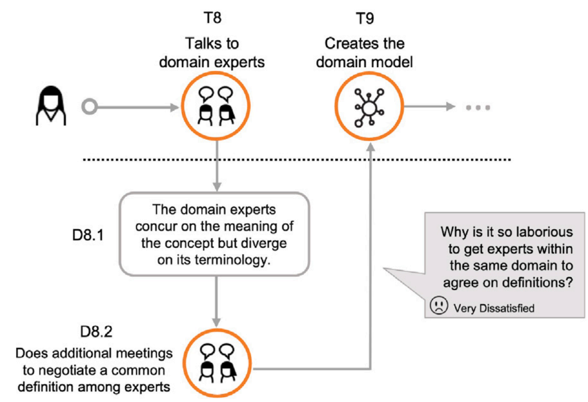

Fig. 7. Example of an occurrence of *PP04 (Effortful Negotiation)*.

models has been questioned. Despite the availability of approaches like Agile Modeling [49] and empirical evidence supporting their benefits [50], many practitioners remain doubtful, arguing that the effort involved in creating these models conflicts with Agile values [51].

While repeated exposure to models can mitigate this pain point, overcoming this resistance toward modeling remains challenging due to the abstract nature of models' value. This challenge is exemplified by one participant in the virtual brainstorming session who wrote: *"I had difficulties explaining to others why I chose a given modeling language and what criteria I used"*. Despite recent contributions on the Return on Modeling Effort (RoME) and the value of modeling [16], comprehensive investigations into cost–benefit analysis in modeling remain scarce in the literature.

PP03 (Expensive Requirements): "It was costly to define the model requirements."

This pain point reflects a modeler's frustration with the financial and cognitive resources required for eliciting model requirements. Unlike the *Improper Requirements* pain point, which concerns requirement quality, this frustration stems from the efforts involved in their definition, regardless of their adequacy. For example, this may arise when multiple stakeholder meetings are needed to reach agreement on model requirements, or when a modeler requires more time than anticipated to translate stakeholders' needs into model specifications, as illustrated in Fig. 6.

Although defining model requirements can be frustrating and resource-intensive, investing time and effort upfront can significantly reduce problems and costs later in the modeling process. Model-driven methodologies in software development help identify defects and misalignments between stakeholders and developers early on, whereas later adjustments tend to be more costly and disruptive [52]. Challenges contributing to this frustration include limited expertise in requirements engineering or modeling, complexities inherent to specific domains or project scopes, and constraints posed by the methods applied. Participants in the virtual brainstorming session experienced these issues firsthand. One participant wrote: *"The abstract nature of conceptual models made it difficult for me to define clear and objective requirements"*. Another one added: *"My lack of domain knowledge made it difficult for me to understand the problem and propose a solution"*.

PP04 (Effortful Negotiation): "It was costly to negotiate a common definition among the experts about a concept of the project's domain of interest."

This pain point is related to a modeler's frustration with the financial and cognitive investments required to determine which aspects of the domain to represent in a model. This process may involve prolonged negotiation with stakeholders to reach an agreement on the model scope, relevant entities and their relationships, and terminology [53]. Such negotiations can drive up the resources required for model development, resulting in additional meetings, potential deadlocks, and increased costs, as noted by Neuhaus et al. (2022) and illustrated in Fig. 7. Nevertheless, it is advisable to invest time in defining concepts at the outset to prevent issues later in the process.

Achieving agreement is a key challenge in modeling, requiring modelers to establish a participatory process and mediate disagreements. However, many modelers lack training in social negotiation skills, as noted in [53] and echoed by a participant in the virtual brainstorming session, who wrote: *"I did not have much experience in extracting domain knowledge from experts"*. Modelers may also have limited expertise in the domain of interest or face difficulties in extracting and formalizing stakeholders' knowledge. Conversely, stakeholders may lack modeling proficiency or struggle to articulate their thoughts and understandings. Domain complexity can further hinder the definition of concepts, leading to uncertainties even among experts, as another participant in the virtual brainstorming noted: *"I felt that even the domain experts did not have reliable knowledge about the domain"*. Effective modeling methods should support concept definition by facilitating the articulation of ontological commitments [54] and providing a rationale for modeling decisions to avoid misunderstandings or false agreements [55].

PP05 (Effortful Verbalization): "It was costly to generate verbalizations of my model suitable for the different audiences that should be able to read and use it."

This pain point concerns the frustration experienced by a modeler because of the financial and cognitive investments required to develop model versions tailored to diverse agents. For example, investments in translating key regularities from a domain model – such as types, attributes, relationships, and constraints – into natural language to facilitate stakeholder understanding. Constraints in verbalization can diminish the model's cost-effectiveness by restricting its usability and reuse, whereas attempting to cater to all potential audiences may lead to unnecessary resource expenditure. Nevertheless, effective verbalization enhances accessibility and understanding, fostering adoption by

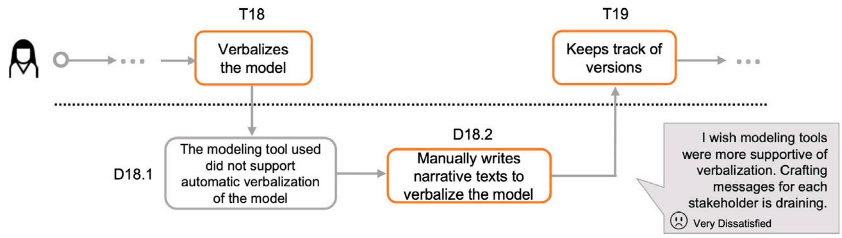

Fig. 8. Example of an occurrence of *PP05 (Effortful Verbalization)*.

non-specialists and bridging communication gaps between modelers and industry professionals [2].

Model verbalization depends on the modeler's ability to define the target audience and select appropriate formats, yet these tasks are not trivial. First, a model created for one purpose may later serve additional contexts, requiring verbalizations that accommodate unforeseen audiences. Second, modeling initiatives typically involve multiple stakeholders, necessitating different verbalization formats tailored to each group. For instance, narrative texts work well for non-technical users, whereas technical glossaries and specifications are suitable for expert audiences. The quality of modeling tools is also crucial to model verbalization, as shown in Fig. 8. Despite advances in tools and methods like the NORMA Software Tool [56], the OntoUML Lightweight Editor (OLED) [57], the semantics of business vocabulary and business rules (SBVR) transformations [58], and the Natural Language for E-R (NaLER) method [59], modelers still report challenges with verbalization, often due to tool limitations and insufficient guidance.

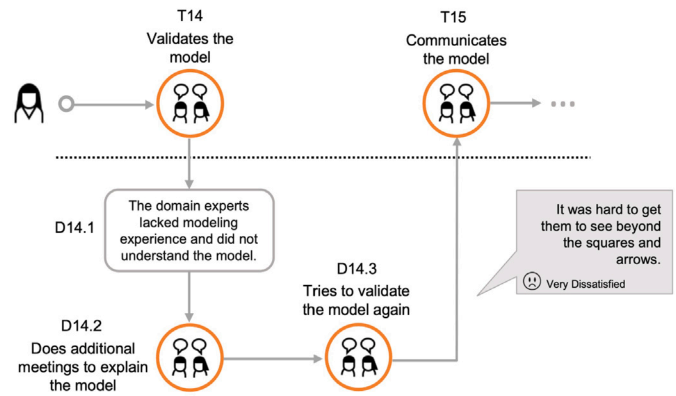

Fig. 9. Example of an occurrence of *PP06 (Inadequate Explanation)*.

PP06 (Inadequate Explanation): "I had difficulty explaining my model to my stakeholders/domain experts."

This pain point reflects the frustration experienced by a modeler when explaining elements of the model – such as classes, attributes, and relations – to team members with limited understanding or interest in modeling. As shown in Fig. 9, this frustration stems from stakeholders' limitations in understanding the explanation and modelers' challenges in providing it. One participant in the virtual brainstorming session wrote about this: *"The other stakeholders did not understand my model, and I had difficulties explaining it"*. In domain modeling, the complexity of the models further complicates communication and understanding for those unfamiliar with the field.

Despite the challenges, as discussed in the *Effortful Verbalization* pain point, helping non-modelers understand models can increase their value by facilitating use and reuse, supporting broader adoption, and ensuring continuity. While structural organization, diagrammatic layout, and documentation can help clarify the model's content, explaining models to diverse stakeholders without structured methodologies can complicate this process. This may lead to delays, rework, and additional financial costs, especially if the communication challenges result in ineffective explanations. Previous studies suggest that, due to these challenges, many modelers avoid explaining their models to team members [2]. Those who do often rely on strategies they have developed through experience, which require skills and expertise that not all modelers possess.

PP07 (Unclear Conceptualization): "Writing definitions for the elements in my model was hard and boring."

This pain point pertains to the frustration experienced by a modeler in crafting definitions for the classes and properties within a model. It involves determining which components require definitions to avoid ambiguity and misinterpretation by users. Unlike the *Effortful Negotiation* pain point, modelers facing this frustration are already familiar with the meaning of the model elements; their challenge is formalizing them. Although no standardized method for defining elements exists, established guidelines for conceptual definitions can help mitigate this pain point.

Since the definitions can vary in form and content depending on a model type and its context [60], writing effective definitions often depends on modelers' expertise. However, even experienced modelers may struggle with complex domains or limited access to domain experts. For instance, participants in a virtual brainstorming session mentioned challenges such as *"It was hard to deal with the complexity and contradictory aspects of the domain"* and *"The domain experts were not available as much as I needed"*. The collaborative nature of defining model elements adds further complexity, particularly when stakeholders lack sufficient modeling knowledge. If these challenges lead to imprecise definitions, delays, and rework may occur in later activities reliant on model clarity, such as implementation and maintenance. An example of this is shown in Fig. 10, where an unclear class definition caused a programmer to misinterpret it, resulting in a flawed implementation and requiring rework.

PP08 (Complexity Issues): "The model has become too complex and too costly to maintain."

This frustration stems from the challenge of determining when complexity reaches a point where the model becomes unmaintainable, and whether techniques to address this issue could and should be applied. It pertains to situations where, even if model maintenance is technically feasible, the effort required may outweigh the benefits. Model complexity, measured by factors such as the number of classes, associations, and attributes, directly correlates with maintainability [61]. Without proper maintenance, however, the model may lose value for some stakeholders due to limitations in its use and reuse. For instance, consider the scenario depicted in Fig. 11, where a model was devised to steer the development of a system. If the model is not maintained alongside system updates, it will become unsuitable for reuse in system maintenance or training new programmers.

Complexity management is often influenced by modelers' expertise and choice of modeling tools. Some tools lack essential features, such

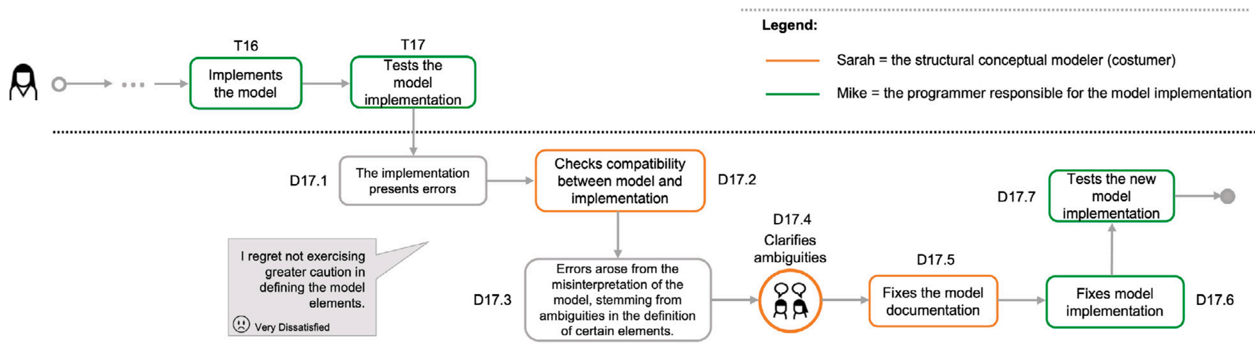

Fig. 10. Example of an occurrence of *PP07 (Unclear Conceptualization)*.

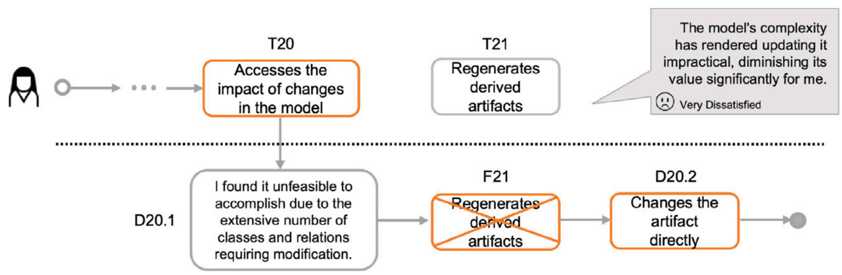

Fig. 11. Example of an occurrence of *PP08 (Complexity Issues)*.

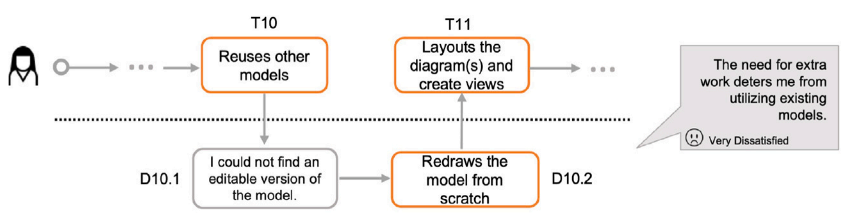

Fig. 12. Example of an occurrence of *PP09 (Laborious Reuse)*.

as support for model abstraction [62], segmentation [63], and clustering [64], which help minimize complexity. In these cases, modelers may use their expertise to find workarounds. Alternatively, they may use tools with these features but lack proficiency in utilizing them. For instance, one participant in the virtual brainstorming session noted: *"I felt there was a lack of formal guidelines for creating model views. I would like to learn strategies to reduce the model's complexity and make it easier to understand"*. In either case, the modeler's ability to address complexity and ensure the model's ongoing usefulness is hindered, leading to frustration in the maintenance process.

PP09 (Laborious Reuse): "I wanted to reuse an existing model, but I had to redraw it from scratch."

This pain point relates to the frustration modelers experience when attempting to build upon existing models without fully recreating them, a challenge that can significantly hinder model reuse. Our research shows that, due to these difficulties, modelers often choose to develop new models rather than adapt existing ones. For instance, one participant in the virtual brainstorming session wrote: *"I tried to reuse an existing model but found it very difficult to adapt it to my needs"*.

Challenges in model reuse stem from issues of model availability, quality, and compatibility. For instance, as shown in Fig. 12, reusing the model without access to the original editable files is laborious. Despite evidence suggesting that model sharing could benefit both modelers and stakeholders by reducing time, cost, and effort [2], modelers are often reluctant to share due to confidentiality or personal reasons. Even when editable models are available, their complexity, disorganization, and application specificity can restrict reuse. Complex or disorganized models are difficult to understand and modify, while application-specific models demand extensive adjustments to fit new contexts. Additionally, compatibility issues with modeling tools or languages – such as limited import functions or version incompatibility – can introduce further obstacles. Proficiency gaps and lack of support may also hinder modelers, deterring effective model reuse.

PP10 (Tool Restrictions): "I could not choose the modeling tool I wanted."

This pain point pertains to the frustration modelers experience when they lack autonomy in selecting a modeling language. It can stem from two sources: the imposition of using a tool with which they are not proficient, or the requirement to use a tool that, although familiar, is considered suboptimal for the project. This frustration can impact model quality and hinder activities that depend on model understanding. For instance, a modeler unfamiliar with a tool's functionalities may struggle to create clear diagrams, compromising the model's comprehensibility. The need to use an unfamiliar or unsatisfactory tool can also increase modeling costs. As depicted in Fig. 13, acquiring proficiency in a new tool incurs costs associated with the learning curve. These costs can be substantial, especially when the tool lacks accessible formal guidance. For instance, one participant in the virtual brainstorming session wrote: *"I found it difficult to use the tool I had chosen. I did not know how to handle its features. It was complicated and had some bugs"*.

Despite the expectation that modeling expertise would grant modelers greater autonomy, real-world project constraints often limit this

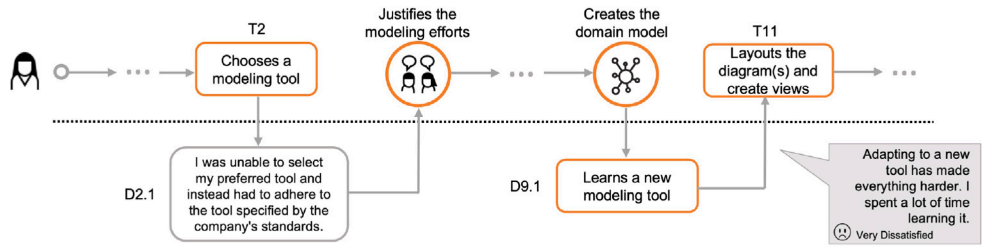

Fig. 13. Example of an occurrence of *PP10 (Tool Restrictions)*.

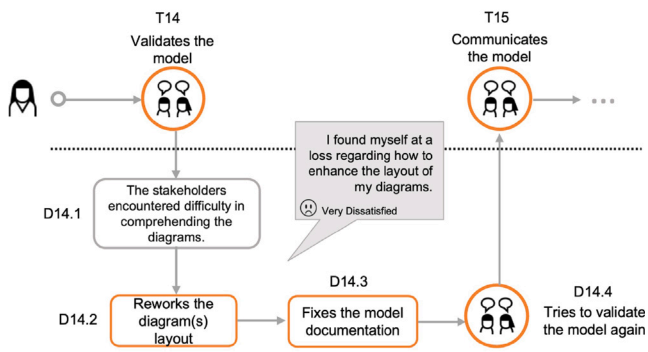

Fig. 14. Example of an occurrence of *PP11 (Effortful Diagramming)*.

freedom. Stakeholders may reject a modeler's preferred tool due to their unfamiliarity with it. Additionally, organizations with established modeling cultures may resist adopting new tools, especially when changes to existing practices are perceived as disruptive. Bureaucratic and scheduling constraints can further limit a modeler's ability to choose the most suitable tool. For instance, one participant in the virtual brainstorming session wrote: *"As licensing professional tools may take time, I had to work with non-ideal resources. Moreover, changing tools during the process was unfeasible due to the timeline"*.

PP11 (Effortful Diagramming): "It was costly to manually split the model into visually appealing and understandable views (diagrams)."

This pain point addresses the frustration modelers experience due to the financial and cognitive resources expended in creating aesthetically pleasing, well-structured, and comprehensible diagrams. While stakeholder preferences for diagram aesthetics vary, research highlights qualities that enhance diagram clarity, such as reducing line crossings, avoiding class overlaps, and standardizing class sizes [65]. Developing effective model views remains a complex task that demands expertise, and even substantial effort does not ensure a satisfactory result. For instance, one participant in the virtual brainstorming session wrote: *"I could not make one of my diagrams look good"*.

A modeler's inability to produce high-quality diagrams may hinder model understanding and usage, contributing to other pain points, such as *Inadequate Explanation* and *Laborious Reuse*. For instance, if stakeholders fail to comprehend the model, they may face challenges in validating it, as depicted in Fig. 14. Inadequate diagrams can increase costs and lead to rework across the modeling process. The challenges in producing quality diagrams may arise from modelers' skills or limitations of the modeling tools. Modelers may lack proficiency in layout design or in selecting essential information for diagrams, or they might use tools lacking layout simplification features. Moreover, even experienced modelers using appropriate tools may struggle with complex domains that inherently produce challenging models.

PP12 (Overlooked Documentation): "My colleagues did not consider it important to document the model and did not understand my efforts to do so."

This pain point reflects a modeler's frustration when their documentation efforts are undervalued or misinterpreted by other project team members, who may perceive it as resource–draining and supplementary, with benefits that are difficult to quantify. As illustrated in Fig. 15, the lack of stakeholder endorsement can impact the modeling process, resulting in delays and increased resource consumption. It can also lead to poorly developed, outdated, or unusable documentation that hinders models' use and understanding, diminishing their value.

Stakeholders undervalue model documentation when it is not aligned with their needs or its role in the modeling process is unclear. This challenge is prevalent in contexts characterized by a limited understanding of modeling and poor organizational modeling practices. Modelers, therefore, must effectively communicate the benefits of documentation to justify their efforts in this task. Proper documentation, for instance, is essential for enhancing model comprehension, usability, and reusability, while potentially reducing specific pain points such as *Effortful Verbalization*, *Inadequate Explanation*, *Unclear Conceptualization*, and *Laborious Reuse*.

PP13 (Limited Verification): "The tool I chose did not support model verification. My model was large and complex, and verifying it myself was not trivial."

This pain point relates to a modeler's frustration with manually verifying whether the syntactic rules of the chosen modeling language have been accurately implemented within the model. For instance, in OntoUML, this verification process would uncover issues such as the incorrect specialization of one "kind" entity by another "kind" entity. Verifying models by hand requires considerable cognitive and financial resources, especially as model complexity increases. An example is the National Agency of Terrestrial Transportation (ANTT) Ontology [66], which includes over 1200 classes and 655 associations. Manually verifying such a large model would be costly and time-consuming, with higher risks of errors. These errors, when discovered in later implementation stages, are harder to correct and may result in extensive rework, which further raises costs [52].

As shown in Fig. 16, this pain point is intensified when modelers use tools lacking automated verification functions, often due to unfamiliarity with available tools or organizational constraints. The latter issue was discussed in the *Tool Restrictions* pain point. Even with appropriate tools, modelers may still experience frustration if they lack expertise in using verification features or are working with complex domains.

PP14 (Unclear Approach): "It was costly to define the model-driven approach."

This pain point reflects the frustration modelers face due to the financial and cognitive demands of defining a model-driven approach. This process involves specifying actions, methodologies, tools, expected

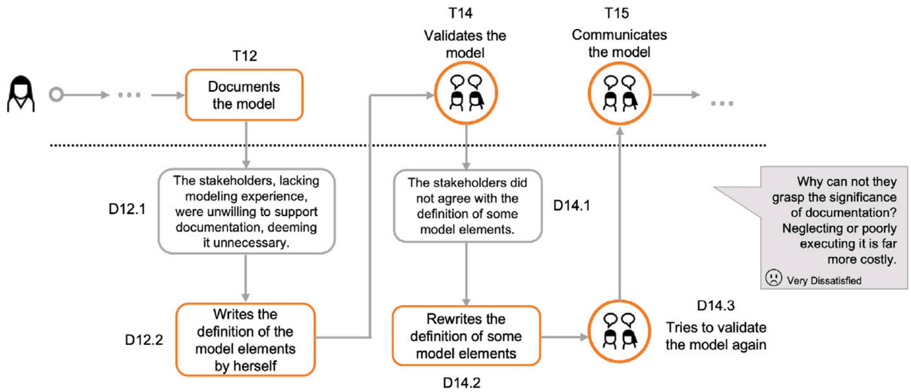

Fig. 15. Example of an occurrence of *PP12 (Overlooked Documentation)*.

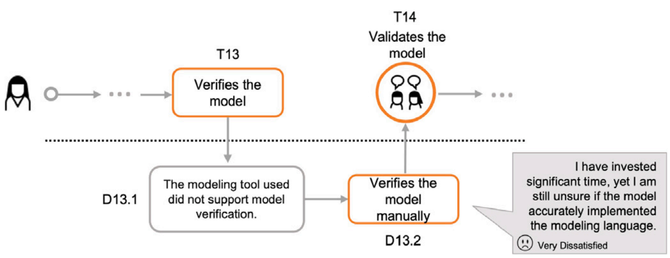

Fig. 16. Example of an occurrence of *PP13 (Limited Verification)*.

outcomes, and roles throughout the modeling process. How these definitions are shaped depends on the modelers' experience, the project's context, and the organizational culture. While over-investing in defining the approach may cause frustrations, it helps avoid the complications of inadequate definitions, such as delays, inefficiencies, and increased resource demands. As illustrated in Fig. 17, the benefits and drawbacks of defining a model-driven approach often become evident indirectly as the modeling process unfolds.

Novice modelers, lacking the insights and preferences that experienced practitioners have developed over time, are particularly susceptible to this frustration. For instance, one participant in the virtual brainstorming session shared: *"It took me a lot of effort to understand the model-driven working methods available and to identify the one that best suited the scope of the model I aimed to create"*. The lack of modeling guidelines within organizations can further contribute to this frustration; as another participant noted: *"I spent a lot of resources on defining the model-driven approach because, for instance, my company did not have a modeling culture"*. The difficulty in finding, adopting, or applying established approaches also presents an issue. These challenges are illustrated in the following statements: *"I could not find a model-driven working method ready to use"*, *"I felt that model-driven methodologies did not stimulate collaboration. I was hoping to find one that allowed for social construction, but I could not"*, and *"I missed more lean and agile working methods for model development. Maybe they exist, but I did not find them"*. Finally, ambiguities in the project scope hinder defining the model-driven approach, as it is hard to design a strategy without clarity on the desired outcomes.

PP15 (Disinterest in Use): "I was the only one on the team interested in using and reusing the model."

This pain point emphasizes the frustration modelers experience when others involved in a project fail to benefit from the model's advantages. This lack of engagement is more prevalent than one might expect. Findings in [2] show that knowledge of a model and its potential for reuse is often confined to the modelers themselves. While this frustration may not directly impact the modeling process, it can affect the cost-benefit of the modeling initiative. Expanding a model's use for its intended purpose or repurposing it for diverse applications allows stakeholders to recognize its benefits, enhancing its perceived value and helping to foster a modeling culture within an organization [2].

During an interview, a modeler shared an example of this frustration when hired by a software company to create a model for documenting its software. As shown in Fig. 18, the company limited the model's use to internal communication among programmers, despite its potential for external application, due to concerns over disclosing proprietary information. This frustration was compounded when the model became outdated and fell into disuse after the modeler's contract ended. These issues may arise from the lack of a well-established modeling culture and the limited modeling expertise within the company, stakeholders' limited understanding of the model or its value, or the modeler's failure to communicate its benefits effectively.

PP16 (Unclear Documentation): "I was not sure how to document my model."

This pain point addresses the frustration modelers experience when uncertain about the optimal approach to document a model. Model documentation involves creating artifacts to enhance understanding (e.g., metadata, visualizations, versioning, concept definitions, verbalization), generating machine-readable annotations, and preparing files accessible as Web resources [67]. Model documentation is complex, as its requirements vary based on the model's type, purpose, and development context. Without standardized frameworks, modelers must decide on suitable documentation approaches, balancing the risk of over-investing in unnecessary details against the potential problems caused by under-documenting. This challenge is reflected in the observation of one participant in the brainstorming session: *"I felt there was a lack of formal guidelines for documenting conceptual models. I was not sure which documents should accompany my model"*.

Despite its complexity, model documentation is essential, as inadequate documentation can hinder model comprehension, leading to delays, operational errors, or increased costs in the modeling process. For instance, as shown in Fig. 19, insufficient clarity about class relationships may force programmers to seek clarification from modelers, delaying system development. Alternatively, they might make arbitrary decisions, risking operational errors in the system, as shown in Fig. 10. Five key factors contribute to uncertainties in model documentation: limited documentation experience, a lack of organizational modeling culture and standardized methodologies, insufficient peer or educational support, and unclear project scopes.

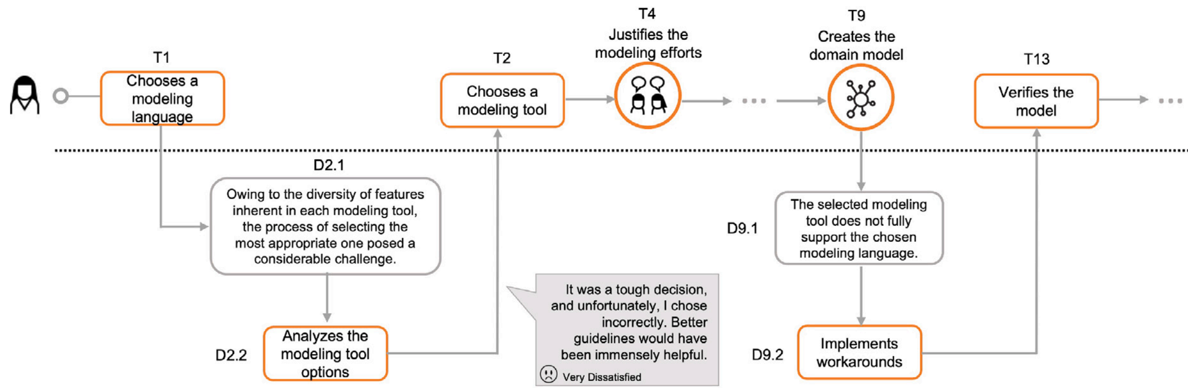

Fig. 17. Example of an occurrence of *PP14 (Unclear Approach)*.

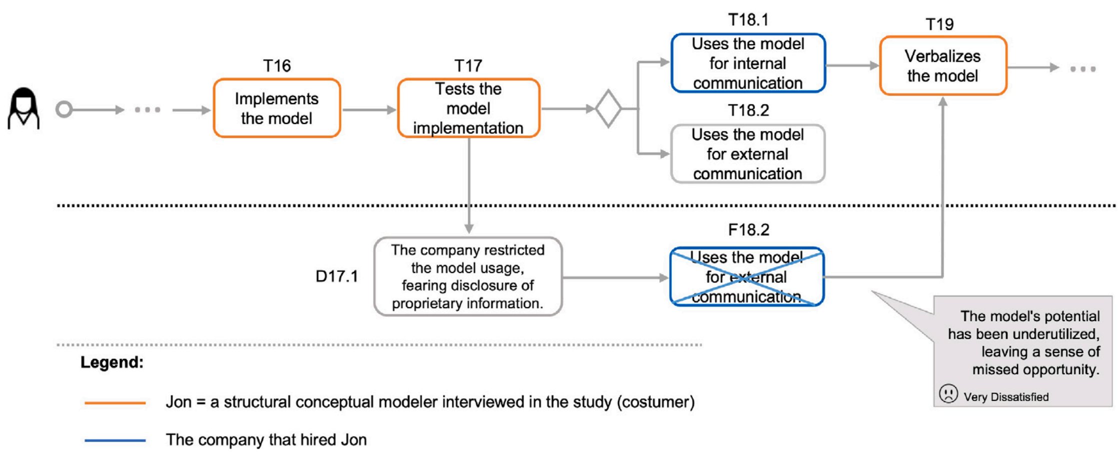

Fig. 18. Example of an occurrence of *PP15 (Disinterest in Use)*.

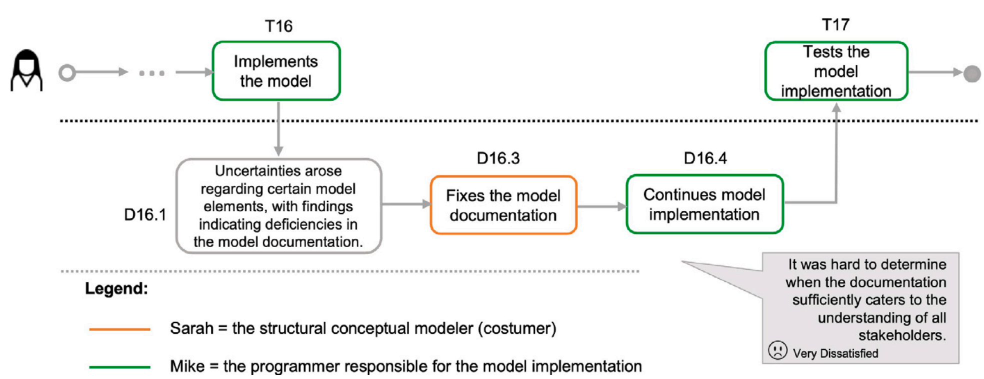

Fig. 19. Example of an occurrence of *PP16 (Unclear Documentation)*.

5. Causes of pain points

Upon comprehending and delineating the prioritized pain points, it became apparent that they could be categorized by certain causal factors, as presented in Table 2. These causal factors were selected based on our view of the modeling process as a dynamic interaction between products, services, and agents. In this perspective, pain points arise when unexpected situations disrupt these interactions, making their causes traceable to specific elements within the modeling process.

Modeler-related Frustrations They may be associated with limited experience among modelers across various domains, reflected in insufficient skills and knowledge. This includes technical difficulties, such as using specific modeling tools and creating visually coherent diagrams, as well as socio-technical challenges, like negotiating shared definitions for model elements. Additionally, human factors, such as personality traits and self-efficacy, may also contribute to these frustrations [68]. As they are linked to individual capabilities, they remain independent of the modeling context or the methodology employed. For example, individuals lacking proficiency in model requirements elicitation may experience similar frustrations when defining competency questions for an ontology or crafting user stories for agile methodologies.

Stakeholders-related Frustrations They may be associated with limited experience among stakeholders across various domains, reflected in insufficient skills and knowledge. This includes technical challenges, such as difficulties in understanding diagrams and notations, as well as socio-technical challenges, like clearly expressing goals or articulating an understanding of concepts.

Table 2
Potential causes of the pain points.

Pain Point	Modeler	Stakeholder	Organization	Method	Tool	Language	Model	Domain	Project
Improper Requirements	X	X		X				X	X
Resistance to Modeling	X	X	X	X					
Expensive Requirements	X	X		X				X	X
Effortful Negotiation	X	X		X				X	
Effortful Verbalization	X				X	X			
Inadequate Explanation	X	X		X			X		
Unclear Conceptualization	X	X		X				X	
Complexity Issues					X			X	
Laborious Reuse					X	X	X		
Tool Restrictions		X	X						
Effortful Diagramming	X				X			X	
Overlooked Documentation	X	X	X						
Limited Verification	X	X	X		X				
Unclear Approach	X		X	X					X
Disinterest in Use	X	X	X						
Unclear Documentation	X		X	X					X

Organization-related Frustrations They may be attributed to the lack of a well-established modeling culture within an organization. This deficiency is characterized by the absence of predefined standards for implementing model-based approaches, such as the selection of modeling languages and tools to be used. It also includes the lack of employee acceptance toward embracing modeling-based solutions.

Method-related Frustrations They may arise from not adopting a method, using the wrong method, misusing a method, or choosing a method that lacks sufficient guidance for its employment. This deficiency in guidance may manifest in the lack of resources such as manuals, instructional materials, discussion forums, an active community, or accessible mentors. The term "method" is used in a broad sense here, yet its interpretation should be contextualized based on the specific pain point to which it addresses. For example, frustrations associated with methods within the context of the *Improper Requirements* pain point are linked to requirements elicitation techniques. In contrast, within the realm of the *Effortful Negotiation* pain point, these frustrations are associated with consensus-building methodologies or negotiation techniques.

Tool-related and Language-related Frustrations They may stem from adopting the wrong modeling tool/language. For instance, this could entail selecting a tool that lacks functions to facilitate modeling activities (e.g., verbalization, verification), or choosing a modeling language that does not support concrete representations due to insufficient domain and comprehensibility appropriateness [69]. Additionally, these frustrations may arise from the improper use of a suitable modeling tool/language or from selecting a modeling tool/language that lacks guidance for its use. This lack of guidance can be analogously interpreted as discussed in the Method-related Frustrations.

Model-related Frustrations They can be attributed to challenges in understanding and accessing the model, which limits modelers' ability to use and reuse it. When ensuring the model accurately reflects the domain, the struggle to comprehend it may be attributed to various factors. For example, issues may stem from an inadequate layout of model diagrams, insufficient model documentation, or the complexity of the domain, which can lead to a complex model structure. Challenges in accessing the model depend on how it was made available. For instance, models partially published, not accompanied by documentation, and without editable files, are difficult to access and reuse.

Domain-related Frustrations They may originate from the domain of interest, whose complexity demands attention from the modeler in selecting appropriate methodologies, tools, and languages for the modeling process. Such frustrations may also arise from difficulties accessing and comprehending domain-specific information, whether provided by domain experts or technical materials such as documentation, regulations, and scholarly works.

Project-related Frustrations They can be ascribed to difficulties stemming from ambiguities within the project scope. Endeavors to articulate the needs of stakeholders as modeling goals are prone to failure when stakeholders themselves lack clarity regarding them.

These broader categories of causes should be analyzed within a specific project contexts to guide the identification of solutions that account for financial, time, personnel, and scope constraints. Many of these pain points can be mitigated through technical solutions by using or improving tools and methodologies. For example, the appropriate use of AI-based tools can help address the *Improper Requirements* pain point by compensating for modelers' lack of skills in requirements engineering or mitigate the *Expensive Requirements* pain point by improving the efficiency of the model requirements definition process [70]. AI can also support the *Unclear Conceptualization* pain point by assisting in defining model elements [71], the *Effortful Diagramming* pain point by evaluating the layout quality of UML class diagrams [65], and the *Laborious Reuse* pain point by extracting information from models that exist only as images [72]. Other pain points may be better addressed through social approaches focusing on improving human skills and knowledge. For instance, gamified learning can support stakeholder engagement in modeling [73], contributing to the mitigation of pain points such as *Resistance to Modeling*, *Inadequate Explanation*, *Overlooked Documentation*, and *Disinterest in Use*. Additionally, it can enhance modelers' preparation for real-world scenarios, helping to address pain points like *Effortful Negotiation*, *Inadequate Explanation*, *Unclear Approach*.

6. Practical implications

The findings of our study have some implications for research and practice, contributing to the ongoing discourse in academic and industrial contexts. The pain points identified in this work highlight key challenges and present valuable opportunities for further investigation. They can be examined collectively or categorized by underlying causes or the specific modelers affected. They can be treated as hypotheses, prompting follow-up empirical studies to generate additional evidence and refine our understanding of modelers' frustrations and their causes.

This study provides researchers with insights into the modeling experience, facilitating the development of methodologies and knowledge that address its challenges. Understanding the pain points associated with existing methods and tools can guide the creation of improved artifacts that meet modelers' needs. Furthermore, by addressing the pain points caused by inexperience among modelers and stakeholders, researchers can develop educational materials and courses to improve modeling training. Such initiatives can broaden the accessibility of modeling practices, attracting new modelers and strengthening the modeling community.

The results can also inform developers and designers of modeling languages and tools about the difficulties users encounter. By understanding and addressing the pain points caused by language and tool issues, they can improve their products and user experiences.

From the modeler's perspective, acknowledging these pain points is critical to planning and decision-making. Recognizing difficulties with tools and languages enables modelers to compare solutions and select the most appropriate one. Understanding pain points related to experience helps modelers identify areas for professional improvement. Thus, our findings contribute to developing more capable modelers who are aware of challenges and equipped to address them.

As discussed, the identified pain points introduce additional costs into the modeling process. These costs may arise from the pain points themselves (e.g., delays, rework) or from efforts to prevent or resolve them (e.g., training, tool development). Therefore, by comprehending these pain points, modelers, organizations, and stakeholders can better strategize their modeling initiatives to avoid or anticipate challenges, making costs more transparent and manageable. It also helps identify and manage the emotions of all participants, positively influencing the initiative's success. Therefore, this study's insights present application opportunities across various contexts and stakeholders.

Although this study frames pain points as sources of frustration, some modelers may perceive certain challenges, such as the *Effortful Negotiation* pain point, as inherent to the modeling process. Furthermore, learning and experience play a crucial role in how modelers respond to these challenges. Pain points that initially cause frustration, such as *Improper Requirements* and *Expensive Requirements*, may become more manageable as modelers gain experience. It is important to recognize that these pain points may or may not arise depending on the individuals involved and the specific context of the modeling process.

7. Threats to validity

The study has four primary vulnerabilities: (i) the potential omission of crucial information that could lead to pertinent pain points, (ii) the risk of misinterpreting data resulting in incorrect pain points, (iii) the possibility of biased prioritization of pain points that may deviate from the reality, and (iv) the possibility for our results to be transferable to other contexts.

The susceptibility to overlooking crucial information was pronounced during Phase 1, in which the goal was to discover as many pain points as possible. One notable concern during this phase was the phenomenon known as the Rosy View Phenomenon, which suggests that individuals tend to provide more positive evaluations of events shortly after they occur [74]. Since all data collection methods relied on modelers' recollections, this may have led to the capture of fewer pain points than actually occurred. Additional concerns include the potential for researcher bias during the thematic analysis, particularly in the coding process. Finally, there was concern that the purposive literature review may have overlooked or excluded relevant works, leading to an unbalanced perspective on the issue [32]. Three strategies were implemented to mitigate the validity threats associated with self-reported data, potential bias in thematic analysis, and the risk of overlooking inputs in the purposive literature review. First, data were gathered from three distinct sources – interviews, academic papers, and modelers' opinions – ensuring a comprehensive approach through data triangulation [75]. Second, the data analysis process was carried out with meticulous attention to detail, ensuring that every expression or word indicating frustration or pain points was thoroughly examined. Additionally, the thematic analysis and the purposive literature review were subject to peer review by multiple researchers to enhance the internal validity of the findings.

Throughout the virtual brainstorming session, while there were no issues concerning the quantity and quality of ideas generated, two concerns arose regarding their length and repetition. Virtual groups generate fewer elaborated ideas than traditional settings, which could explain the issue of brevity [76]. Additionally, the use of idea displays in virtual environments may impede participants' ability to recall previously expressed concepts, leading to repeated ideas. To address this limitation, the first author reviewed all Post-it notes on the Mural platform and conducted individual follow-up discussions with each participant. This approach helped minimize moderator bias and ensured more thorough idea exploration. Furthermore, to prevent the recurrence of ideas, the study incorporated a synthesis stage following the brainstorming session.

Misinterpretation of data posed a significant risk during the synthesis process in Phase 2. While triangulating different methods can mitigate the limitations of individual approaches, combining qualitative methods can raise concerns about the transparency of data sources and the validity of interpretations [33]. Moreover, researchers must be aware of personal biases when synthesizing qualitative data, as reciprocal translations aimed at providing a holistic view may still introduce bias. To minimize these risks, all data were traceable, and the synthesis outcomes were independently validated by two researchers.

Biased prioritization of pain points was a concern during Phase 4, with potential biases arising from participants and mediator [15]. To address participant bias, Delphi rounds were conducted anonymously and separately, with data randomly organized. Moreover, participant selection was aligned with the study's scope, based on experience with modeling and relevant contextual factors. Several factors support the validity of our approach: (i) while the sample size was small, it adhered to the recommendations of research methodologists [15]; (ii) the sample consisted exclusively of professionals actively engaged in modeling, aligning with our focus on understanding modelers' pain points; and (iii) although gender representation was imbalanced, we assert that gender did not significantly affect modelers' frustrations. To mitigate mediator bias, we used quantitative measures for synthesizing results and leveraged the automated feedback features of the Mural platform. As a result, mediator intervention was minimal, primarily limited to clarifications regarding the task or platform use.

Finally, we assessed the potential for our findings to be applicable in contexts beyond those of the modelers and organizations involved in this study. Since empirical findings may be transferable based on contextual similarities [75], we considered whether similar results might emerge if modelers from different organizations with comparable contexts were included. The frustrations experienced by our participants could be shared by others in organizations that lack a conceptual modeling culture, face inadequate tool support, or struggle with decisions about modeling languages and tools. While we hypothesize that practitioners in such organizations might encounter similar challenges, we recognize the need for future research to validate this hypothesis. Further studies are required to explore the specific contextual factors that influence the emergence of pain points and identify potential solutions. Only through additional empirical research in real-world contexts can we gather the necessary evidence to assess the generalizability of our findings, thus providing a direction for future research.

8. Related work

To the best of our knowledge, no previous literature evaluated pain points in domain modeling in the manner undertaken in our study. Nevertheless, several works have examined pain points, difficulties, challenges, and pitfalls within modeling-related contexts. This section highlights the most relevant contributions.

Authors in [77] explored the challenges and barriers in model-based software engineering (MBSE), identifying issues that align with our findings, including model accuracy, lack of verification and validation, and tool-related difficulties. Although their study relies on experiential insights rather than a structured methodology, it offers valuable best practices for addressing these challenges.

While not all difficulties translate into pain points, several studies analyzing challenges in using and developing models reveal parallels with our findings. For instance, two resent studys [4,78] examined modeling difficulties in conceptual data model construction, comparing novice and proficient modelers using an adapted Entity-Relationship Model. They identified cognitive breakdowns, such as issues with relationship types, generalization hierarchies, and entity-relationship classification. These challenges align with the frustrations we observed in categorizing domain aspects according to language specifications, indicating their findings as a subset of ours.

The study presented in [79] identified four barriers to conceptual model use through a survey of 304 practitioners, employing a similar coding process. Three of these barriers – costs, uncertainties, and tool-related complexities – overlap with the pain points identified in our study. The fourth barrier, concerning the proliferation of modeling methods developed by practitioners, can be explained by our findings on the *Unclear Approach* pain point, which highlights modelers' difficulty in finding suitable methods. The study also outlined success factors in conceptual modeling, offering insights for future research.

In [80], the authors conducted a meta-review of 21 papers on Model-Driven Engineering (MDE), 15 of which addressed practitioners' difficulties. The most common issue was the lack of essential tool functionalities, followed by challenges related to organizational culture, methodological maturity, and the effort needed to acquire modeling skills. While the study had a broader focus, many of the nine key issues in MDE highlighted align with our research findings.

Several studies [81–84] have investigated the challenges students face when using Unified Modeling Language (UML), identifying issues that resonate with our findings. For instance, they found difficulties related to modelers' lack of understanding of domain constraints and language constructs, uncertainty regarding model correctness, insecurity regarding abstracting concepts, unfamiliarity with modeling software, and deficiencies in UML support materials. Although these studies focused on novice modelers and the model creation phase, they addressed a broader range of scenarios. Some also offered recommendations to mitigate these challenges.

Similar investigations in business process modeling have produced analogous findings. For instance, issues such as stakeholder engagement, translating business goals into technical objectives, and role discrepancies within project frameworks were identified in [85]. These challenges reflect our findings on pain points stemming from causes related to stakeholders and projects. Similarly, challenges related to the lack of standardization in modeling notations, tools, and methodologies, as well as difficulties in demonstrating the value of process modeling were highlighted in [86]. These factors also contribute to the frustrations in our study.

In two of his works [87,88], Rosemann identified challenges in process modeling through focus groups and semi-structured interviews with analysts and managers. The research methods used align with ours, and many of the pitfalls noted by the author, such as lack of synergy, qualified modelers, and model maintenance, correspond to causes of pain points in our study. Despite focusing on a different modeling type, the author's work is significant for examining challenges across the entire modeling process.

Finally, in [89], the authors investigated the variation in symbol sets across different modeling notations, focusing on perceptual discriminability, visual expressiveness, and semantic transparency. Using four sets of symbols in visual process modeling languages, the authors collected data from 136 participants engaged in modeling tasks. The study found that notational deficiencies concerning perceptual discriminability and semiotic clarity impact comprehension, cognitive load, and the time needed to understand models. This aligns with our finding that suboptimal choices of modeling language and tool, and poor visual presentation contribute to modelers' frustration.

9. Conclusion

This study aimed to identify and analyze the main pain points experienced by modelers engaged in structured conceptual modeling. To achieve our goal, we conducted a five-phase empirical study using a multi-method approach. Initially, we identified 71 distinct pain points, which were synthesized into a condensed list of 41. Next, we prioritized 16 pain points as the most significant and prevalent in domain modeling. Finally, we refined, documented, and exemplified each of them, analyzed their potential causes, and discussed their practical implications.

Our findings contribute to the body of knowledge in conceptual modeling by bringing empirical evidence that sheds light on what it takes to transfer modeling methods into organizations. In this context, we provide valuable insight for organizations, modelers, and other stakeholders. When adopting a conceptual modeling approach, organizations may consider our work as a foundation for cost–benefit analyses. Therefore, they can leverage our findings to foster a more conducive modeling culture and enhance the success rates of their modeling initiatives. Modelers will find valuable insights for improving their experience and optimizing the modeling process in several critical ways. These insights can facilitate more effective planning of modeling initiatives; provide insights for evaluating modeling costs; suggest alternative solutions to overcome modeling difficulties; contribute to the refinement of modeling methods, languages, and tools; and provide guidance for the professional development of novice modelers. Moreover, they can support modelers in enhancing their planning and decision-making regarding the modeling process and their efforts toward technical self-improvement. Stakeholders in various roles can leverage these insights to understand how to manage potentially frustrating situations, creating a more positive environment and improving overall project success. Thus, addressing these pain points is critical to optimizing the modeling process and fostering continuous improvement in modeling practices.

Continuing within the domain of conceptual modeling, our findings provide valuable resources for advancing various aspects of the field. They support researchers in refining modeling methods, languages, and training approaches. Tool developers may also benefit from the findings to enhance their products. Moreover, our study contributes to the discourse on the reusability of conceptual models. By providing empirical evidence, we support the argument for making conceptual models reusable and for keeping them reusable over time. We also advocate for sharing these models with the broader community beyond the walls of individual organizations.

Our article also advances the development of several related fields, the broader domain of model-driven software engineering and requirements engineering in particular. Today, research into AI-based systems increases both in-depth and breadth, and conceptual models are employed to ensure these systems are meaningful to people. Understanding the pains associated with developing these models is an important part of developing the knowledge of how to create and employ them cost-effectively. Additionally, within the field of empirical software engineering, our work represents a pioneering effort, to the best of our knowledge, in employing a multi-method research design in an exploratory context in the conceptual modeling area. We demonstrate the complementary use of a qualitative interview data analysis method, a survey research method, and the Delphi method, to assure depth and breadth in our exploration of the phenomenon of interest.

Future research efforts could delve deeper into understanding the pain points identified in this study and explore potential solutions. Investigating the behavior of these pain points in a broader sample would also be a valuable research direction. It may allow the identification of patterns related to factors such as the experience level of the modeler, the type of modeling language used, and the dynamics among project stakeholders. There is also a need for future studies to examine further the relationship between these pain points and modeling costs, aiming to establish clearer and potentially quantifiable relationships. In addition, we recommend that future research consider distinctions between development methodologies, such as Agile and conventional approaches, to better contextualize the influence of modeling practices on the experience of these pain points.

CRediT authorship contribution statement

Isadora Valle: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Tiago Prince Sales:** Validation, Supervision, Methodology, Conceptualization. **Eduardo Guerra:** Writing – review & editing, Supervision, Methodology. **Maya Daneva:** Writing – review & editing, Methodology. **Renata Guizzardi:** Writing – review & editing. **Luiz Olavo Bonino da Silva Santos:** Writing – review & editing. **Hend*erik* A. Proper:** Writing – review & editing, Conceptualization. **Giancarlo Guizzardi:** Writing – review & editing, Supervision, Conceptualization.

Declaration of Generative AI and AI-assisted technologies in the writing process

While preparing this work, the authors used ChatGPT to improve the readability and language of the manuscript. After using this tool, the authors reviewed and edited the content as needed, taking full responsibility for the content of the published article.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Study findings and replication materials are available in the replication package [31].

References

[1] J. Mylopoulos, Conceptual modelling and telos, Concept. Model. Databases CASE (1992) 49–68.
[2] I. Valle, et al., What do I get from modeling? An empirical study on using structural conceptual models., in: 27th EDOC, 14367, Springer, 2023, pp. 21–38.
[3] G. Guizzardi, H.A. Proper, On understanding the value of domain modeling, in: Proceedings 5th VMBO, 2835, CEUR-WS.org, 2021, pp. 51–62.
[4] K. Rosenthal, et al., Modeling difficulties in data modeling: Similarities and differences between experienced and non-experienced modelers, in: ER, Springer, 2020, pp. 501–511.
[5] B. Wang, et al., A biclustering-based method for market segmentation using customer pain points, Eng. Aappl. Artif. Intel. 47 (2016) 101–109.
[6] I. Wiese, et al., Naming the pain in developing scientific software, IEEE Softw. 37 (2020) 75–82.
[7] S. Chattopadhyay, et al., What's wrong with computational notebooks? Pain points, needs, and design opportunities, in: Proceedings of the 2020 CHI, 2020, pp. 1–12.
[8] D.M. Fernández, et al., Naming the pain in requirements engineering: Contemporary problems, causes, and effects in practice, Empir. Softw. Eng. 22 (2017) 2298–2338.
[9] J. Sithambaram, et al., Issues and challenges impacting the successful management of agile-hybrid projects: A grounded theory approach, Int. J. Proj. Manage. 39 (5) (2021) 474–495.
[10] R.J. Schonberger, Frustration-driven process improvement, Bus. Horiz. 61 (2018) 297–307.
[11] D. Graziotin, et al., On the unhappiness of software developers, in: Proceedings 21st EASE, 2017, pp. 324–333.
[12] V. Braun, V. Clarke, Using thematic analysis in psychology, Qual. Res. Psychol. 3 (2006) 77–101.
[13] C. Wilson, Brainstorming and Beyond: A User-Centered Design Method, Newnes, 2013.
[14] R. Huelin, et al., What's in a name? Systematic and non-systematic literature reviews, and why the distinction matters, Evidence (2015) 34–37.
[15] M.M. Grime, G. Wright, Delphi method, Wiley Statsref: Stat. Ref. Online 1 (2016) 16.
[16] H.A. Proper, G. Guizzardi, Modeling for enterprises; let's go to RoME ViA RiME, in: The Practice of Enterprise Modeling Forum, 2022.
[17] N. Guarino, et al., On the philosophical foundations of conceptual models, in: Information Modelling and Knowledge Bases XXXI, IOS Press, 2020, pp. 1–15.
[18] E.S.K. Yu, Social modeling and i*, in: A. Borgida, et al. (Eds.), Conceptual Modeling: Foundations and Applications - Essays in Honor of John Mylopoulos, Vol. 5600, Springer, 2009, pp. 99–121.
[19] M. Chinosi, A. Trombetta, BPMN: An introduction to the standard, Comput. Stand. Interfaces 34 (2012) 124–134.
[20] J. Gordijn, J.M. Akkermans, Value-based requirements engineering: exploring innovative e-commerce ideas, Requir. Eng. 8 (2003) 114–134.
[21] The Open Group, The ArchiMate Enterprise Architecture Modeling Language, Technical Report ArchiMate 3.1 Specification, The Open Group, 2022.
[22] G. Guizzardi, et al., Towards ontological foundations for conceptual modeling: The unified foundational ontology (UFO) story, Appl. Ontol. 10 (2015) 259–271.
[23] P.P.-S. Chen, The entity-relationship model: Toward a unified view of data, ACM Trans. Database Syst. 1 (1976) 9–36.
[24] OMG, UML 2.0 Superstructure Specification – Final Adopted Specification, Technical Report ptc/03-08-02, Object Management Group, 2003.
[25] T.A. Halpin, Object-role modeling (ORM/NIAM), in: Handbook on Architectures of Information Systems, Springer, 2006, pp. 81–103.
[26] J. Ruiz, et al., Evaluating user interface generation approaches: model-based versus model-driven development, Softw. Syst. Model. 18 (2019) 2753–2776.
[27] R. Halvorsrud, et al., Involving users in the development of a modeling language for customer journeys, Softw. Syst. Model. 22 (2023) 1589–1618.
[28] B.D. Temkin, Mapping the customer journey, Forrester Res. 3 (2010) 20.
[29] N. Johns, What is this thing called service? Eur. J. Mark. 33 (1999) 958–974.
[30] S. Easterbrook, et al., Selecting empirical methods for software engineering research, in: Guide To Advanced Empirical Software Engineering, Springer, 2008, pp. 285–311.
[31] I. Valle, et al., Replication package, 2024, https://github.com/utwente-scs/pain-points-modeling.
[32] D.A. Cook, Systematic and nonsystematic reviews: Choosing an approach, Heal. Simul. Res.: A Pr. Guid. (2019) 55–60.
[33] L.A. Jensen, M.N. Allen, Meta-synthesis of qualitative findings, Qual. Health Res. 6 (4) (1996) 553–560.
[34] R.A. Falbo, SABiO: Systematic approach for building ontologies, in: 1st Joint Workshop ONTO.COM / ODISE on Ontologies in Conceptual Modeling and Information Systems Engineering, Vol. 1301, CEUR-WS.org, 2014.
[35] A. Gómez-Pérez, M.C. Suárez-Figueroa, Neon methodology for building ontology networks: a scenario-based methodology, in: Proceedings S3T 2009, Universidad de Sofia, 2009.
[36] J.-J.V. Wintraecken, The NIAM Information Analysis Method: Theory and Practice, Springer Science & Business Media, 2012.
[37] P. Kruchten, The Rational Unified Process: An Introduction, Addison-Wesley Professional, 2004.
[38] A. Tong, et al., Consolidated criteria for reporting qualitative research (COREQ): a 32-item checklist for interviews and focus groups, Int. J. Qual. Health Care 19 (2007) 349–357.
[39] C. Wohlin, Guidelines for snowballing in systematic literature studies and a replication in software engineering, in: Proceedings 18th EASE, ACM, 2014, pp. 1–10.
[40] P. Nandwani, R. Verma, A review on sentiment analysis and emotion detection from text, Soc. Netw. Anal. Min. 11 (1) (2021) 81.
[41] C. Strapparava, A. Valitutti, WordNet-affect taxonomy - SKOS version, 2013, (Accessed: June 2023).
[42] Philip Lief Group, Roget's 21st century thesaurus, 2013, (Accessed: June 2023).
[43] R. House, Dictionaryċom unabridged, 2023, (Accessed: June 2023).
[44] C. Powell, The Delphi technique: myths and realities, J. Adv. Nurs. 41 (4) (2003) 376–382.
[45] L.S. Fish, D.M. Busby, The delphi method, Res. Methods Fam. Ther. 469 (1996) 482.
[46] D. Firesmith, Specifying good requirements, J. Object Technol. 2 (2003) 77–87.
[47] G.K.Q. Monfardini, et al., Use of competency questions in ontology engineering: A survey, in: Conceptual Modeling, Springer, 2023, pp. 45–64.
[48] C. Mettke-Hofmann, Neophobia, in: Encyclopedia of Animal Cognition and Behavior, Springer, 2022, pp. 4582–4590.
[49] S. Ambler, Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process, Wiley, 2002.
[50] A. Gupta, et al., Conceptual models in agile software development: A possible solution to requirements engineering challenges in agile projects, IEEE Access 10 (2022) 119745–119766.
[51] K.M. Beck, et al., Manifesto for agile software development, 2001, Online.
[52] S.W. Liddle, Model-driven software development, in: Handbook of Conceptual Modeling: Theory, Practice, and Research Challenges, Springer, 2011, pp. 17–54.

[53] F. Neuhaus, J. Hastings, Ontology development is consensus creation, not (merely) representation, Appl. Ontol. 17 (2022) 1–19.
[54] G. Guizzardi, F. Baião, M. Lopes, R. Falbo, The role of foundational ontologies for domain ontology engineering: An industrial case study in the domain of oil and gas exploration and production, Int. J. Inf. Syst. Model. Design (IJISMD) 1 (2) (2010) 1–22.
[55] N. Guarino, Formal ontology in information systems, in: Proceedings 1st FOIS, Vol. 46, IOS Press, 1998.
[56] M. Curland, T.A. Halpin, The NORMA software tool for ORM 2, in: Information Systems Evolution – CAiSE Forum 2010, Vol. 72, Springer, 2010, pp. 190–204.
[57] J. Guerson, et al., OntoUML lightweight editor: A model-based environment to build, evaluate and implement reference ontologies, in: 19th EDOCW, IEEE, 2015, pp. 144–147.
[58] J. Cabot, R. Pau, R. Raventós, From UML/OCL to SBVR specifications: A challenging transformation, Inf. Syst. 35 (2010) 417–440.
[59] C. Atkins, J. Patrick, NaLER: a natural language method for interpreting entity-relationship models, Campus-Wide Inf. Syst. 17 (2000) 85–93.
[60] S. Michie, et al., Creating ontological definitions for use in science, Qeios (2019).
[61] M. Genero, et al., Assessing object-oriented conceptual models maintainability, in: Advanced Conceptual Modeling Techniques: ER 2002 Workshops, ECDM, MobIMod, IWCMQ, and ECOMO, Springer, 2003, pp. 288–299.
[62] A. Egyed, Semantic abstraction rules for class diagrams, in: Proceedings ASE 2000, IEEE, 2000, pp. 301–304.
[63] G. Figueiredo, et al., Breaking into pieces: An ontological approach to conceptual model complexity management, in: 12th RCIS, IEEE, 2018, pp. 1–10.
[64] J. Akoka, I. Comyn-Wattiau, Entity-relationship and object-oriented model automatic clustering, Data Knowl. Eng. 20 (2) (1996) 87–117.
[65] G. Bergström, et al., Evaluating the layout quality of UML class diagrams using machine learning, J. Syst. Softw. 192 (2022) 111413.
[66] C.A.M. Bastos, et al., Building up a model for management information and knowledge: The case-study for a Brazilian regulatory agency, in: 2nd International Workshop on Software Knowledge, 2011, pp. 3–11.
[67] D. Garijo, WIDOCO: A wizard for documenting ontologies, in: The Semantic Web – ISWC 2017, Springer, 2017, pp. 94–102.
[68] S. Meliá, et al., The impact of personality and self-efficacy on domain modeling productivity in graphical and textual notations, Inf. Softw. Technol. 173 (2024) 107491.
[69] G. Guizzardi, et al., An ontology-based approach for evaluating the domain appropriateness and comprehensibility appropriateness of modeling languages, in: MODELS, Springer, 2005, pp. 691–705.
[70] K. Ronanki, et al., Requirements engineering using generative ai: Prompts and prompting patterns, in: Generative AI for Effective Software Development, Springer, 2024, pp. 109–127.
[71] G.P. Rees, R. Lew, The effectiveness of OpenAI GPT-generated definitions versus definitions from an English learners' dictionary in a lexically orientated reading task, Int. J. Lexicogr. 37 (2024) 50–74.
[72] F. Chen, et al., Automatically recognizing the semantic elements from UML class diagram images, J. Syst. Softw. 193 (2022) 111431.
[73] N. Pflanzl, Gamification for Business Process Modeling, Westfaelische Wilhelms-Universitaet Muenster (Germany), 2018.
[74] T.R. Mitchell, et al., Temporal adjustments in the evaluation of events: The "rosy view", J. Exp. Soc. Psychol. 33 (1997) 421–448.
[75] R. Wieringa, Design Science Methodology for Information Systems and Software Engineering, Springer, 2014.
[76] R. Ziegler, et al., Idea production in nominal and virtual groups: Does computer-mediated communication improve group brainstorming? Group Process Intergroup Relat. 3 (2000) 141–158.
[77] B. Papke, et al., MBSE model management pain points–wait, this looks familiar!, in: INCOSE International Symposium, Wiley Online Library, 2023, pp. 273–289.
[78] K. Rosenthal, S. Strecker, Toward a taxonomy of modeling difficulties: A multimodal study on individual modeling processes, EMISA Forum 41 (2021) 31–32.
[79] P. Fettke, How conceptual modeling is used, Commun. Assoc. Inf. Syst. 25 (2009) 43.
[80] C. Verbruggen, M. Snoeck, Practitioners' experiences with model-driven engineering: A meta-review, Softw. Syst. Model. 22 (2023) 111–129.
[81] K. Siau, P.-P. Loo, Identifying difficulties in learning UML, Inf. Syst. Manag. 23 (3) (2006).
[82] V.Y. Sien, An investigation of difficulties experienced by students developing unified modelling language (UML) class and sequence diagrams, Comput. Sci. Educ. 21 (2011) 317–342.
[83] H. Kruus, et al., Teaching modeling in SysML/UML and problems encountered, in: 25th EAEEIE Annual Conference, IEEE, 2014, pp. 33–36.
[84] R. Reuter, et al., Insights in students' problems during UML modeling, in: EDUCON, IEEE, 2020, pp. 592–600.
[85] Y. Alotaibi, Business process modelling challenges and solutions: a literature review, J. Intell. Manuf. 27 (2016) 701–723.
[86] M. Indulska, et al., Business process modeling: Current issues and future challenges, in: Advanced Information Systems Engineering: 21st CAiSE, Springer, 2009, pp. 501–514.
[87] M. Rosemann, Potential pitfalls of process modeling: part A, Bus. Process Manag. J. 12 (2006) 249–254.
[88] M. Rosemann, Potential pitfalls of process modeling: part B, Bus. Process Manag. J. 12 (2006) 377–384.
[89] K. Figl, et al., The influence of notational deficiencies on process model comprehension, J. Assoc. Inf. Syst. 14 (2013) 312–338.