# Empirically Evaluating a Domain Specific Modeling Language for Social Services from the Modeler's Perspective

Felix Holz[1]

[1]Universität Rostock, Albert-Einstein-Straße 22, 18059, Rostock, Germany

**Abstract**

A domain Specific Modeling Language (DSML) promises to to be more suitable for the domain's needs. However, it must also provide support for the modeler, who may have little insight into the domain. This article describes and conducts an empirical study that compares a domain-specific language for modeling flexible and knowledge-intensive social work processes with a general language of the declarative paradigm. The main focus of the study is to examine the usefulness of the language for the modeler. The results show that DSML performs better than the general language in terms of perceived ease of use and comprehensibility.

**Keywords**

Domain Specific Modeling Language, DSML, Social Services, Evaluation

## 1. Introduction

One of the central goals of enterprise modeling is to model processes in order to control workflows, but also to formally store knowledge about working procedures. This is particularly relevant for social service companies, as the work is weakly-structured, highly flexible, and knowledge-intensive [1, 2, 3]. The knowledge intensity is contradicted by high employee turnover and thus the loss of knowledge carriers, forcing the institutions to retrain new staff to these formally implicit known processes. Compared to routine processes modeled with imperative languages, declarative modeling languages like the Case Management Model and Notation (CMMN) [4] are more suited to model knowledge-intensive processes [5, 6]. In the past, however, weaknesses have emerged in the practical application of these modeling languages in the domain of Social Services [1]. On the one hand, CMMN is indeed usable for depicting the casework on a general level but only partially corresponds to the specific needs of the domain, especially regarding the core work like client interaction [1]. Consequently, an approach to design a Domain Specific Modeling Language (DSML), more targeted to the focus group, promises to be more suitable for the domain's needs [7]. We provided a DSML in the past [2, 8] in order to collect the implicit process knowledge of the social workers for analyzing, reusing, and explicating process patterns.

In this contribution, an empirical study is presented that seeks to validate the DSML regarding the usefulness from the modeler's perspective. Therefore, the domain specific language is compared to a general language utilizing the declarative paradigm, namely CMMN. The goal is to investigate whether the DSML is better suited to model the use cases of the domain. The results of the study show tendencies to confirm this claim. Before describing the study and the results in section 3, the DSML for Social Services is briefly introduced in section 2.

*PoEM2025: Companion Proceedings of the 18th IFIP Working Conference on the Practice of Enterprise Modeling: PoEM Forum, Doctoral Consortium, Business Case and Tool Forum, Workshops, December 3-5, 2025, Geneva, Switzerland*
felix.holz@uni-rostock.de (F. Holz)
0000-0002-0877-7063 (F. Holz)

## 2. Previous Work on Domain Specific Modeling for Social Services

In previous works, we introduced the DSML for Social Services and referred to it as the Social Service Notation (SSN) [2]. This section briefly summarizes the history of the SSN's creation. It begins with highlighting existing literature on modeling knowledge-intensive and flexible work. Subsequently, findings from practical experience are condensed, leading to a formal definition of the language. Finally, a practical application example is provided.

### 2.1. Modeling Knowledge-Intensive Processes

Social service providers are considered part of personal services, which are performed on and with a client to achieve the long-term goal of improving the client's mind or body [9]. The core work of Personal Services exhibits the characteristics of knowledge-intensive processes [10]. Knowledge-intensive processes are characterized by both unpredictability and high knowledge intensity, i.e., the process course and outcome depend on the actor's tacit knowledge [11]. According to Boissier et al. [12] and Işik et al. [13], the process's success is not apparent before and after execution. This is caused by highly individual processes for different client requests. Consequently, social workers must bring their experience and knowledge to the work process to achieve given, usually long-term goals under uncertain conditions through working in close cooperation with an individual client while considering their situation and context [3].

Mertens et al. state that such dynamic, non-standard, and knowledge-intensive processes require the run-time flexibility of declarative process modeling [10]. In the declarative modeling paradigm, the execution of any process step is allowed at any time, as long as it is not constrained [14, 5]. CMMN [4] and ConDec [14] are the better-known representatives of this paradigm, but are not considered suitable for the Social Service domain [8, 2, 1].

Sadiq et al. refer to flexibility as the ability of a workflow process to be executed from a "loose" or only partially specified model; every process instance (i.e., case) is unique [15]. Schonenberg et al. [16] define four ways to achieve flexibility in process representations, whereas Reichert and Weber [17] state four requirements for the performance of flexible process specifications. As stated in previous work [2], the most important criterion for Social Services is "flexibility by underspecification" or "looseness", i.e., a free process specification that also allows unplanned activities at run-time [17, 8].

### 2.2. Creating a DSML for Social Services

Based on several expert workshops with various representatives from the domain and the characteristics of knowledge-intensive processes, we have identified three key concepts that are important for working with clients. Social workers often encounter unforeseen situations while interacting with clients and must use their knowledge and experience to determine the necessary course of action. In doing so, they must work toward the client's long-term goal. However, it is also possible that actions are taken to achieve a short-term goal. Thus, the following three core aspects form the basis for representing the knowledge-intensive core work of a social worker: (1) (Client) Goals to be achieved, (2) Situations experienced by the client, and (3) Actions that the social worker can take. The modeling language's models are intended to provide actors with improved work support and work structuredness, as well as an opportunity to externalize process knowledge.

In [8], the SSN was formally defined, methodically based on Frank's [7] guidelines for creating DSMLs. The guidelines range from determining generic and specific requirements to creating a domain vocabulary and formal language specification, up to optionally developing a modeling tool, and finally evaluating the language. A language specification defines syntax, semantics, and graphical notation [18]. One way of formally (and graphically) representing the syntax is with a meta model[19], which was specified in [8], and can be seen in Figure 1.

At the meta model's bottom, the instantiated core concepts are depicted. These core concepts are used for actual modeling and can be divided into `Model Elements` and `Relations`. A `Model Element` (such as `Goal`, `Action`, or `Situation`) can be associated with a `Relation` that involves one or more other `Model elements`. As a specialization of the `Relation` type, the `Connector` relates any two `Model Elements` with each other, modeled by the `Prerequisite` and `Recommend` instantiations. The `Affiliation` relation inherits from the abstract `Composition` type, allowing the modeler to build a hierarchy. Only `Container` `Model Elements` (instantiated by a `Goal`) can be a `Composition` of further `Model Elements`. Henceforth, as a specialization of the `Container`, a `Goal` can contain further `Goals`, `Actions`, and `Situations`. On the other hand, these composed Model Elements belong to exactly this `Goal`.
Semantically speaking, the `Affiliation` relation is intended to give the corresponding `Model Elements` the context in which they are placed in. Like in the declarative paradigm [14], the `Affiliation` represents that an `Action` can be carried out at any time, if it corresponds to the affiliated `Goal`, unless there is a `Prerequisite` to be met. The `Prerequisite` connector does not mean that one process step must necessarily follow another (but certainly can). While the Prerequisite and Recommend relations can be used syntactically the same way, semantically, the former implies a strong, obligatory binding between model elements (i.e., it *has to* be done before). In contrast, the latter softens this relation to show a noteworthy association. (e.g., after doing homework, sleeping is recommended. However, it is possible to sleep without doing homework before.) [8]

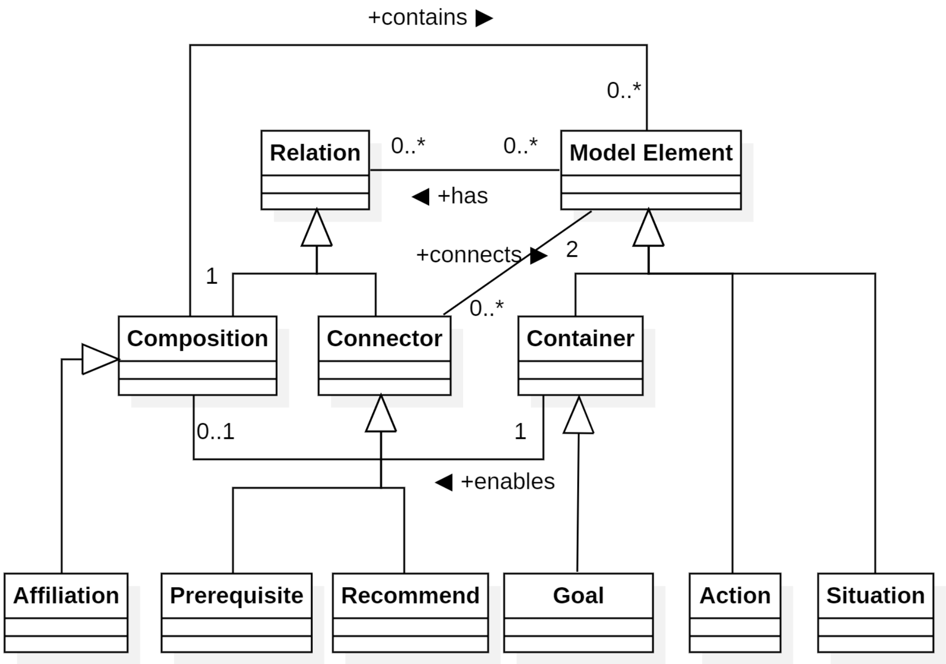


**Figure 1:** Meta model of the Social Service Notation (SSN) showing the syntactical relations between modeling concepts. Source: [8]

## 2.3. Modeling Example

In [8], the SSN was applied in practice. We conducted three workshops with domain experts from a German social institution in the setting of their "Clean Living" project, involving approximately 36 clients. In the workshops, we structured the information collection regarding the client interaction processes in conjunction with the goals of the client cases, gathering subgoals, situations, actions, and recurring patterns. This collected information was refined and transferred into models.
After the models had been presented, they were evaluated in terms of comprehensibility, complexity, correctness, and usability by asking the domain experts (who were primarily involved in the modeling

process) to rate them on a scale of 1-5 according to these criteria. Furthermore, the participants were asked qualitative questions about the benefits they would see in the models. They stated that they considered them helpful in familiarizing themselves with cases, for neutral self-monitoring, and as an aid to orientation by providing an overview of the entire scope of action. They said that it could serve as a guide for action, as it provides rough assistance and serves as a reminder without prescribing strict guidelines, which is particularly useful for "counteracting gut decisions." They also saw potential in supporting self-reflection and examining complex situations. However, they also said that it was not suitable for directly supporting everyday work, since they will not consult the model during client interaction. They stated that it would be used more as a passive source of knowledge and for planning support [8].

An example of a sub model created during these workshops can be found in Figure 2. For demonstrating the SSN in practice, some of the model's statements are highlighted and explained in the following, derived from [8]:

1. The client's main `Goal` depicted in this model is *abstinence*, and it has the sub goal for *preventing addiction pressure*. This goal has further sub goals to divide the means for prevention into short-, mid-, and long-term actions. They may not appear as classical goals but more like work phases. A **goal hierarchy** is built with the nested `Affiliation` relations to provide better structure and an overview of which measures can be taken depending on time criticality.
2. Acting as a start-situation, if a new client has arrived with the goal of *abstinence*, it has to be clarified how the client deals with the addiction and the reasons for it. Note that *clarification*, compared with *conversation*, has a more binding and committing means to it. This fact is implicitly known but not explicitly modeled. Before the social worker can begin with the *networking* action, the aforementioned actions have to be carried out, i.e., they are a `Prerequisite` for this action.
3. A `Situation` like *acute addiction pressure* can also be a `Prerequisite` for an `Action`. In this instance, the action for *assessing the addiction pressure* is only necessary (or can only be carried out) when the social worker is aware of the situation of *acute addiction pressure*. The awareness of how the client is coping with it (either successful or with recidivism) is only possible after the assessment. Depending on success, further actions are to be carried out.
4. `Actions` like *Control addiction pressure* or *on-call service* do not have any `Prerequisites`, meaning they can be carried out whenever the corresponding `Goal` is active for the client.
5. The `Recommended` relation in this model is mainly depicted around *acute addiction pressure*, meaning first, if the pressure is perceived, it is recommended to initiate the `Goal` for *prevention*. On the other hand, it is recommended to have worked on means for prevention before the addiction pressure arises. Using `Prerequisite` instead of `Recommend` connectors would result in a deadlock. *Listlessness* can result from the addiction pressure, which is an issue for *long-term prevention* actions.

Furthermore, the more abstract wording on the actions allows social workers to perform these tasks appropriately with freedom, based on their personal experience.

## 3. Language Evaluation Through Comparison

Generally speaking, a modeling language has two target groups: on the one hand, the model recipients, i.e., social workers or the domain experts. In the previously mentioned workshops (Section 2.3), the applicability and usefulness of the SSN in the domain of social services were confirmed, as the language produced results that were perceived as useful by domain experts. Furthermore, the language provided modelers with guidelines to structure the collection of knowledge-intensive and flexible domain processes. This can be conceived as the language's validation from the perspective of the model recipients.

On the other hand, a modeling language also serves method experts — mostly people from outside the field with modeling experience and the goal of creating a structure and overview of the domain.

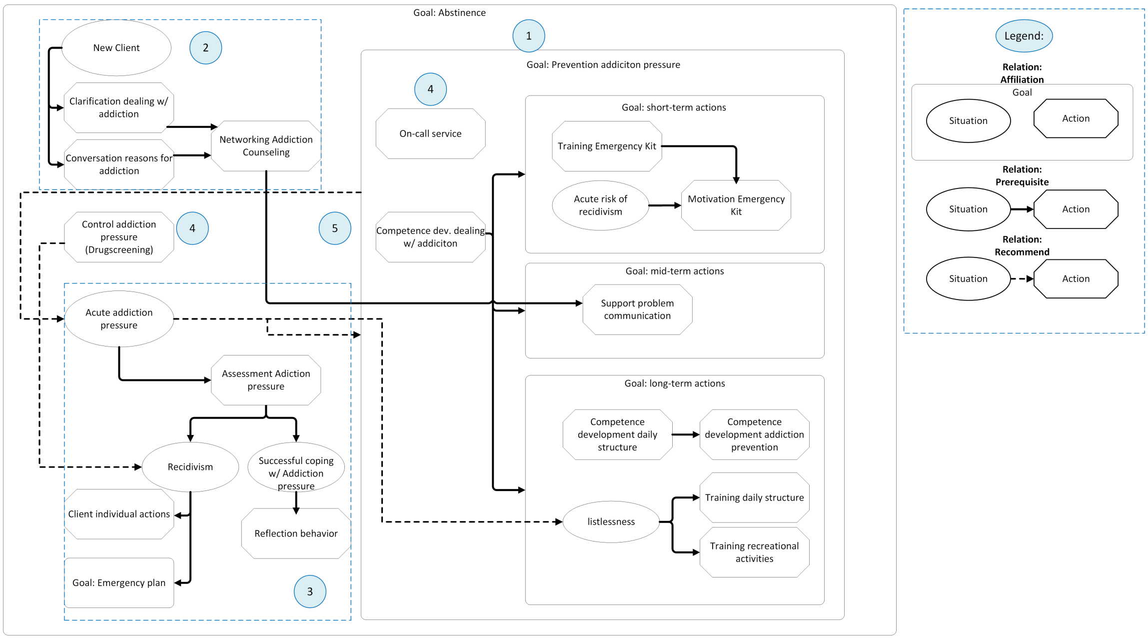


**Figure 2:** SSN model created during expert workshops, depicting occuring situations, actions to be taken, and sub goals to be achieved while working on the goal of the client's abstinence. Source: [8]

To evaluate the language's applicability from the perspective of method experts, an empirical study was conducted, where modelers use the language(s) for knowledge-intensive processes. Since CMMN inspires the SSN but claims to be simpler, more applicable, and easier to understand in the domain of social services [8], it seems reasonable to compare these two languages for validation purposes. In the remainder of this section, literature on modeling language and model evaluation is highlighted before the study design is presented, and the results are evaluated.

### 3.1. Related Work on Evaluating Models and Modeling Languages

SEQUAL is a well-established framework for evaluating the quality of both models and modeling languages [18, p. 205 ff.]. It divides quality criteria into seven dimensions. However, it has been criticized that the criteria cannot be quantified due to vague descriptions [20]. In [20], a several quantifiable and countable characteristics of model quality, along with metrics for measuring them, are specified for process models. Completeness, correctness, relevance, and flexibility thus describe the quality of the model's semantics, while multiple "violations" of various kinds are given for syntax quality. Comprehensibility and unambiguity are measured for the pragmatics of the model. The so-called 3QM framework thus offers many countable quality criteria, but does not sufficiently consider the perceived comprehensibility of the model on a cognitive level.
However, since the SSN is designed to express domain concepts to domain experts and more intuitive knowledge workers with no experience in modeling [8], the language is better evaluated in terms of cognitive capabilities. Moody's "Physics of Notation" is often cited for designing and evaluating graphical representations [21]. This includes nine guidelines for the appropriate visualization and design of modeling languages to enable a cognitively effective transfer of the model's information. Aranda et al. [22] address comprehensibility of model representations, e.g., through "External Cognition" or the Cognitive Dimensions Framework. They also provide guidelines for empirically measuring comprehensibility. However, the focus here lies on the interpreter of the model, not on the modeler and the perception of language quality. In [23], several dimensions for measuring quality are specified, including the Perceived Quality of Modeling Language (PQML), the Perceived Usefulness of the Modeling Procedure (PUMP), and the Perceived Quality of the End Products (PQEP), among others. The respective dimensions to be surveyed focus on the perception of the test subjects and address important aspects of

the modeling process. In addition, the measured concepts are manageable in number, since “for each additional perspective an evaluator considers, the practicality of performing evaluations decreases.” [22]
Additionally, empirical studies already conducted on the perceived quality of modeling languages for knowledge-intensive processes should also be considered. For example, Routis et al. [24] conducted a study to evaluate CMMN (more precisely, the adoption of CMMN), in which process modelers modeled human-centric real-world scenarios. The models were evaluated based on complexity and expressiveness. The user experience was assessed through a discussion and a questionnaire, which was completed after modeling, and contained items on usefulness, ease of use, and usability factors. For the evaluation, they used a multi-criteria decision-making method (Analytic Hierarchy Process).

### 3.2. Empirical Study Design

The empirical study design is aimed at the modelers. The hypothesis to be investigated in the study is: *The SSN is better suited for modeling a use case from the social service domain than CMMN.* To support this hypothesis, the following supporting theses were outlined:

1. **Simplicity, comprehensibility of language use:** The modeling task is simpler and easier to understand with SSN compared to CMMN, measured in terms of perceived ease of use.
2. **Simplicity of modeling:** Modeling with SSN is less time-consuming, measured in the time required by participants to complete the modeling task.
3. **Correctness:** Models created with the SSN depict the domain specific facts more accurately, measured by the rate of syntax and semantic errors.
4. **Complexity of models:** Models created with the SSN are simpler/easier, measured by the number of model elements.

To investigate this hypothesis, the test subjects are given a modeling task and are divided into either a treatment group or a control group (as in [24]), in which the used languages differ. The modeling task has been designed so that the test subjects receive a written scenario. This scenario is based on the model in Figure 2. The key concepts to be modeled were highlighted in the textual description, and the desired relationships between them were presented but not explicitly described (since the relationships between CMMN and SSN differ). Before modeling, the test subjects were introduced to the language within their respective groups. To measure the modeling procedure’s “efficiency” [23], the modeling time was measured for each individual.
After finishing the modeling task, the test subjects are given a questionnaire to determine their perception of the modeling experience. Quantitative questions on understandability, simplicity/ease of use, usefulness, completeness, usability, and satisfaction were asked on a 6-point Likert scale to allow for comparability between the groups. The questions and concepts to be measured were primarily inspired by PQML for evaluating the modeling language and by PUMP for evaluating the modeling procedure [23]. The concepts of simplicity/ease of use were derived from “clarity”, completeness from “conceptual minimalism”, and usability from “Effectiveness” of the modeling procedure. Furthermore, usability and ease of use were concepts also asked in [24]. The focus of this study, however, lies on comprehensibility and ease of use, so multiple questions are used to measure these concepts. The questions are listed in Appendix A.
For the evaluation of the produced models, the goal was to utilize metrics that were as simple and quantifiable as possible to enable comparison. Model size based on the number of model elements and processing time are relatively simple metrics for determining the complexity of the modeling process [24, 25]. For determining model quality, first, the number of semantic and syntactic errors is used to estimate the correctness and complexity. This approach is inspired by counting the number of “violations” in [20]. Second, completeness is measured by showing whether all highlighted elements of the scenario description can also be found in the model.

**Table 1**
Assessment of the modeling process through questionnaires.

| Category | no. Questions | ØCMMN (n=8) | ØSSN (n=8) | p |
|---|---|---|---|---|
| Age | 1 | 27.4 | 25.5 | 0.721 |
| Experience: modeling languages | 1 | 4.875 | 5.125 | 0.6824 |
| Experience: modeling languages | 1 | 4.875 | 5.125 | 0.6824 |
| Understandig presentation | 3 | 4.208 | 5.125 | 0.3299 |
| Understanding task | 3 | 4.666 | 5.167 | 0.4732 |
| Understandability | 3 | 4.208 | 4.708 | 0.3956 |
| Simplicity / Ease of Use | 3 | 3.583 | 4.458 | 0.0264 |
| Completeness | 1 | 3.875 | 4.75 | 0.287 |
| Usefulness | 2 | 4.375 | 4.688 | 0.4598 |
| Usage | 1 | 3.875 | 4.5 | 0.2867 |
| Satisfaction | 3 | 4.083 | 4.04 | 0.9038 |

### 3.3. Empirical Results

The study was conducted with a total of 16 participants. These were either German students of business informatics or employees of the Chair of Business Information Systems in Rostock. These participants represent method experts, as modeling is a central aspect of their studies. Business informatics specialists play an interdisciplinary role across several domains, and the use of a DSML is precisely intended for them. Modeling experience, therefore, also varies among the participants. They were randomly divided into groups, ensuring that the ratio of students and employees was approximately equal. The introduction of the different languages was recorded in advance by the same person in order to minimize the presenter's influence. The modeling task was carried out with pen and paper to minimize the influence of a modeling tool and to identify syntax errors. Since not every aspect of CMMN was important for the modeling task, concepts such as milestones and discretionary items were omitted from the explanation.

Table 1 shows the results of the questionnaire after modeling was done. Note that the scale ranges from 1 to 6, where 1 indicates complete disagreement and 6 indicates complete agreement. The p-values depicted were computed by utilizing the tool ux-calc [26], which promises to work on comparison tasks with small sample sizes and the Wilcoxon rank sum test, which examines significance between two independent samples [27, pp. 111] [28]. It can be seen that the age and modeling experience of the SSN group is slightly higher on average. This may impact the comprehensibility of the presentation in which the language was introduced. The lower part of the table lists the questions on the modeling experience itself, with the SSN being rated better on average, especially on the focused criteria, understandability, and ease of use. In terms of simplicity/ease of use, even a significant difference was identified (on significance level of $\alpha = 0.05$). CMMN appeared to be significantly less easy to use than the SSN. As already described in preliminary studies [29, 2, 8], the assumption reinforces that there are too many aspects of the language irrelevant to the use case to be modeled. However, in terms of perceived satisfaction with the language, no difference can be seen. The participants perceive both languages as equally helpful in supporting the modeling process.

However, the evaluation of the models and their quality were less straightforward and insightful. The average time required to complete the modeling task was 41.75 minutes for the CMMN group and 43.875 minutes for the SSN group. The maximum completion time was set to 45 minutes. The average number of modeled elements (model elements and relations) was also similar (CMMN=46.7, SSN=45.75).

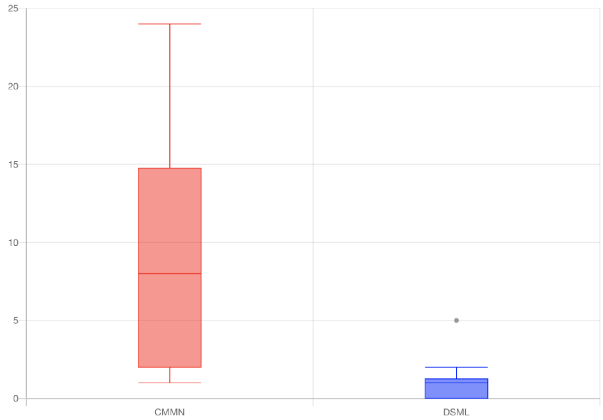


**Figure 3:** Syntax errors in comparison.

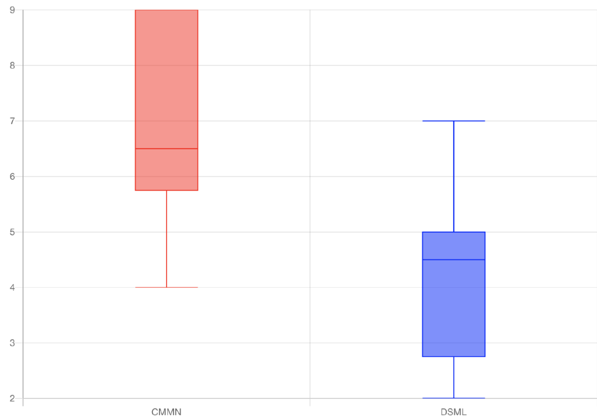


**Figure 4:** Semantic errors in comparison.

Therefore, no conclusions can be drawn regarding complexity. However, it was noticeable that the subjects in the CMMN group mapped fewer of the concepts highlighted in the scenario (19 out of 27 on average) to the model than those in the SSN group (23.5 out of 27). This reinforces the assumption that SSN guides the modeling process more.
The models were independently checked for syntax and semantic errors by three people. Figure 3 shows a comparison of the **syntax errors** found in the two modeling languages. The test subjects made hardly any syntactic errors when using SSN, which is not surprising given that the language allows more flexibility. Similarly, it is easier to make syntactic errors in CMMN, as the set of rules for its use is simply larger, and more model elements can be used. This is reflected in the wide range of errors through the CMMN group's participants. The distribution of **Semantic errors** is depicted in Figure 4. They included, for example, made-up model elements, incorrect interrelationships, or the use of invalid model elements for a concept from the scenario. Here, the SSN has also a slight advantage. Syntax errors are easier to identify and classify than the more subjective semantic errors.

## 4. Conclusion

In summary, the hypothesis *"The SSN is better suited for modeling a use case from the social service domain than CMMN"* can be confirmed. The questionnaire's average values on the language's perceived quality often highlight a slight advantage through several elicited concepts for the SSN; in terms of simplicity/ease of use, the SSN performs significantly better than CMMN (confirming supporting thesis 1). Furthermore, it was clearly evident that the number of syntactic errors found in the SSN models is significantly lower than in the CMMN models (confirming supporting thesis 3). However, the time

spent modeling and the number of model elements used differed only marginally, so that supporting these 2 and 4 could not be confirmed.
The findings have limitations. The number of participants was rather low. Another point of criticism can also be found in the study design itself, which appears to be biased in favor of the SSN. The scenario to be modeled was based on a model that was created with the SSN in mind. However, it is not straightforward to design a clear, realistic scenario for non-domain experts without adapting it into a concise format. It should also be noted that the metric for model quality based on the number of recognized syntax and semantic errors lacks a certain degree of objectivity. CMMN is more formally defined and developed than the SSN; therefore, more potential for error exists. In hindsight, this approach seems less fitting for quantitatively measuring the quality of language comprehension. However, it depicts a fairly simple approach by omitting the interpretation of the models. However, it must be noted that the questionnaire was quite effective in producing usable findings. The items selected reflected the purpose of the survey well.
Furthermore, a criteria-based, argumentative-deductive comparison between CMMN and SSN is planned to validate the language, incorporating, for example, the SEQUAL framework [18], Moody's Physics of Notation[21], or quality criteria from meta models [30]. However, it is worth noting that many of the criteria and frameworks listed in the literature place a high value on software engineering formalisms. Since the language is intended to facilitate interdisciplinary communication, a selection of criteria relevant to the purpose of the language must first be made. Another case for future studies might be an empirical evaluation of model comprehensibility [22].

## Acknowledgments

I would like to thank Karl Eichenmüller, Niklas Götz and Nico Blasek; the students, who operationally carried out the study. Further, I would like to thank Michael Fellmann for supervising me and my PhD thesis.

## Declaration on Generative AI

During the preparation of this work, the author used DeepL, Grammarly in order to: Grammar and spelling check, rephrasing, text translation. After using this tool/service, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

## References

[1] P. Herzog, B. Lantow, Adaptive case management in social institutions [adaptive case management in sozialen einrichtungen], Lecture Notes in Informatics (LNI), Proceedings - Series of the Gesellschaft fur Informatik (GI) 275 (2017). URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85083245918&doi=10.18420%2fin2017_81&partnerID=40&md5=5643c046a0966913361641a0aebf78c8.
[2] F. Holz, M. Fellmann, B. Lantow, Ein modellierungskonzept zur prozessstrukturierung für soziale dienstleister, in: M. Riebisch, M. Tropmann-Frick (Eds.), Modellierung 2022, Gesellschaft für Informatik e.V, Bonn, 2022, pp. 171–180. doi:10.18420/modellierung2022-015.
[3] B. Lantow, T. Baudis, F. Lambusch, Mining personal service processes, in: W. Abramowicz, R. Corchuelo (Eds.), Business Information Systems Workshops, volume 373 of *Springer eBook Collection*, Springer International Publishing and Imprint Springer, Cham, 2019, pp. 61–72.
[4] OMG, Case management model and notation, version 1.0. technical report may, omg, may 2014, 2014.
[5] S. Goedertier, J. Vanthienen, F. Caron, Declarative business process modelling: principles and modelling languages, Enterprise Information Systems 9 (2015) 161–185.

[6] S. Haarmann, A. Seidel, M. Weske, Modeling objectives of knowledge workers, in: A. Marrella, B. Weber, Marrella (Eds.), Business Process Management Workshops, volume 436 of *Lecture Notes in Business Information Processing*, [s.n.], [S.l.], 2022, pp. 337–348.
[7] U. Frank, Domain-specific modeling languages: Requirements analysis and design guidelines, in: I. Reinhartz-Berger, A. Sturm, T. Clark, S. Cohen, J. Bettin (Eds.), Domain engineering, Springer, Berlin and Heidelberg, 2013, pp. 133–157.
[8] F. Holz, D. Vogel, M. Fellmann, Specification and application of a domain specific modeling language for social services, in: INFORMATIK 2023 - Designing Futures: Zukünfte gestalten, Gesellschaft für Informatik e.V., Bonn, 2023, pp. 1879–1894. doi:10.18420/inf2023_190.
[9] P. Halmos, The personal service society, The British Journal of Sociology 18 (1967) 13.
[10] S. Mertens, F. Gailly, G. Poels, Enhancing declarative process models with dmn decision logic, in: S. Nurcan, S. Guerreiro, Q. Ma, R. Schmidt (Eds.), Enterprise, Business-Process and Information Systems Modeling, volume 214 of *Springer eBook Collection Computer Science*, Springer, Cham, 2015, pp. 151–165.
[11] M. Szelagowski, A. Lupeikiene, Business process management systems: Evolution and development trends, Informatica (Netherlands) 31 (2020) 579–595. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85092910703&doi=10.15388%2f20-INFOR429&partnerID=40&md5=73b50057fd1ab1b6c1885d3b6987e582.
[12] F. Boissier, I. Rychkova, B. Le Grand, Challenges in knowledge intensive process management, Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOCW 2019-October (2019). URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85076030824&doi=10.1109%2fEDOCW.2019.00023&partnerID=40&md5=63439069a160f77223d107971d3a6885.
[13] Ö. Işik, W. Mertens, J. van den Bergh, Practices of knowledge intensive process management: quantitative insights, Business Process Management Journal 19 (2013) 515–534.
[14] M. Pesic, W. M. P. van der Aalst, A declarative approach for flexible business processes management, in: J. Eder, S. Dustdar (Eds.), Business process management workshops, volume 4103 of *Lecture Notes in Computer Science*, Springer, Berlin, 2006, pp. 169–180.
[15] S. Sadiq, W. Sadiq, M. Orlowska, Pockets of flexibility in workflow specification, in: H. S. Kunii (Ed.), Conceptual modeling, volume 2224 of *Lecture Notes in Computer Science*, Springer, Berlin, 2001, pp. 513–526.
[16] Schonenberg, M.H., Mans, R.S., Russell, N.C. , Mulyar, N.A., Aalst, van der, W.M.P., Towards a taxonomy of process flexibility (extended version), BPM reports, BPMcenter. org, 2007.
[17] M. Reichert, B. Weber, Enabling flexibility in process-aware information systems: challenges, methods, technologies, Springer Science & Business Media, 2012.
[18] J. Krogstie, Model-based development and evolution of information systems: A Quality Approach, Springer Science & Business Media, 2012.
[19] D. Bork, D. Karagiannis, B. Pittl, A survey of modeling language specification techniques, Information Systems 87 (2020) 101425.
[20] S. Overhage, D. Q. Birkmeier, S. Schlauderer, Qualitätsmerkmale,-metriken und-messverfahren für geschäftsprozessmodelle, Wirtschaftsinformatik 54 (2012) 217–235.
[21] D. Moody, The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering, IEEE Transactions on Software Engineering 35 (2009) 756–779.
[22] J. Aranda, N. Ernst, J. Horkoff, S. Easterbrook, A framework for empirical evaluation of model comprehensibility, in: 2007 International Workshop on Modeling in Software Engineering, IEEE, Piscataway, NJ, 2007, p. 7.
[23] D. Ssebuggwawo, S. Hoppenbrouwers, E. Proper, Assessing collaborative modeling quality based on modeling artifacts, in: W. van der Aalst, J. Mylopoulos, N. M. Sadeh, M. J. Shaw, C. Szyperski, P. van Bommel, S. Hoppenbrouwers, S. Overbeek, E. Proper, J. Barjis (Eds.), Practice of Enterprise Modeling: Third IFIP WG 8.1 Working Conference, PoEM 2010, Delft, The Netherlands, November 9-10, 2010. Proceedings, volume 68 of *Lecture Notes in Business Information Processing*, Scholars Portal, Berlin, Heidelberg, 2010, pp. 76–90. doi:10.1007/978-3-642-16782-9{\textunderscore}6.

[24] I. Routis, C. Bardaki, G. Dede, M. Nikolaidou, T. Kamalakis, D. Anagnostopoulos, Cmmn evaluation: the modelers' perceptions of the main notation elements, Software and Systems Modeling 20 (2021) 2089–2109.
[25] M. A. Marin, H. Lotriet, J. A. van der Poll, Metrics for the case management modeling and notation (cmmn) specification, in: R. J. Barnett (Ed.), Proceedings of the 2015 Annual Research Conference on South African Institute of Computer Scientists and Information Technologists, ACM Digital Library, ACM, New York, NY, 2015, pp. 1–10.
[26] M. F. A. A. F. C. P.-M. L. Shang-Lin Chen, Chantale Labbé, Ux calculator [computer software]., 2020. HEC Montréal, Montréal, Canada. Retrieved from https://uxcalc.web.app/».
[27] R. V. Hogg, E. A. Tanis, D. L. Zimmerman, Probability and statistical inference, volume 993, Macmillan New York, 1977.
[28] C. Ford, The wilcoxon rank sum test., 2017. Https://virginia.edu/data/articles/the-wilcoxon-rank-sum-test.
[29] B. Lantow, M. Dehne, F. Holz, Evaluating notations for product-service modeling in 4em: general concept modeling vs. specific language (2019).
[30] Z. Ma, X. He, C. Liu, Assessing the quality of metamodels, Frontiers of Computer Science 7 (2013) 558–570.

## A. Questions

In the following the questions of the questionnaire are stated, originally asked in German:
**Understandability:**

- The presented modeling language is understandable.
- The purpose of the given model elements is easy to understand.
- The possible relationships between the model elements are understandable.

**Ease of Use / Simplicity:**

- It is easy to use the modeling language. / The modeling language is easy to apply.
- Modeling with the presented modeling language is easy.
- The complexity of the modeling language has made modeling difficult. (invert Scale)

**Usefulness**

- The modeling language is useful.
- The given model elements were useful to model the facts from the scenario.

**Completeness:**

- The given model elements were sufficient to model the facts from the scenario.

**Usability/Usage:**

- Overall, I felt that the modeling language helped me to model the given facts.

**Satisfaction:**

- Overall, I am satisfied with the model I created.
- Overall, I am satisfied with how the modeling process went.
- I am satisfied with the modeling language.

**Further Usage:**

- I will continue to use the modeling language.
- I will recommend the modeling language to others.