





A Systematic Comparison of Case Management Languages

Julia Holz¹, Luise Pufahl²(✉) , and Ingo Weber² 

¹ Technische Universität Berlin, Berlin, Germany
j.holz@campus.tu-berlin.de

² Technische Universität Berlin, Software and Business Engineering, Berlin, Germany
{luise.pufahl,ingo.weber}@tu-berlin.de

Abstract. Our world and many industries, such as healthcare or consulting, are becoming more digital and focused on knowledge workers. In consequence, flexible, knowledge-intensive business processes are increasingly relevant for organizations, and subject to increased interest. Case management languages support the management of knowledge-intensive processes with respect to documentation, analysis, execution, and monitoring. Besides the standard Case Management Model and Notation (CMMN), other languages have been developed, such as fragment-based Case Management (fCM) and PHILharmonic Flows. So far, no structured comparison of their functionalities and applicability has been undertaken. With this work, we compare CMMN, fCM, and PHILharmonic flow with two structured methods: a functional comparison and a user study. We found that fCM offers broad functionality, but CMMN was perceived to be easier for users.

Keywords: Case management · Comparison · User study

1 Introduction

With an increased automation of structured business processes, knowledge-intensive processes (KIPs) receive growing attention by business organizations [37]. KIPs are driven by knowledge workers who use their expertise and experience to drive a case based on its characteristics. Such processes are often emergent, knowledge and goal-oriented, event-driven, and possibly constraint and rule-driven [6]. Similar to structured business processes, knowledge-intensive processes need to be managed throughout the whole BPM life-cycle [18].

Case management refers to a process management approach that can support the flexible nature of knowledge-intensive processes. It provides concepts, methods, and techniques to manage KIPs' need for variability, adaptation, information, and compliance. Since the first approach to case management was developed by Van der Aalst et al. [34], several alternatives have been designed in research. *Data-oriented* languages, such as PHILharmonic Flows (PHIL) [15], capture mainly the relevant data objects of a case and their life cycle and center the modeling and execution of a case around it. *Constraint-oriented* languages,

such as the DECLARE language [25] or DCR (Dynamic Condition Response) graphs [31], focus on the constraints of a case. The knowledge worker is allowed to do everything as long as the constraints and rules are fulfilled. Finally, *stage-oriented* languages, such as fCM (fragment-based case management) [10] as well as the industry standard CMMN (Case Management Model and Notation) [23] divide a case into different stages that can be flexibly combined at runtime.

This broad range of languages led to the challenge for practitioners to select the “right” language for managing their knowledge-intensive processes. Thus, this paper aims to systematically compare case management languages. For this purpose, a selective set of case management modeling languages are investigated from a functional and understandability perspective. Thereby, we focus on two purposes: modeling and documenting KIPs.

In this work, we selected the industry-standard CMMN [23] and compare it to a representative of the data-oriented language, PHIL [1, 15], and a representative of the stage-oriented language, fCM¹ [8, 10], which both provide a modeling language and an execution engine for cases. Constraint-oriented languages are not considered in this work, because they use the declarative modeling approach. The understandability of declarative vs. imperative modeling was already studied in [25] and it was found that imperative modeling languages are more comprehensible. Declarative languages require a certain familiarity with the constructs to achieve user understandability [25]. However, in this work, we plan a user study with a limited training phase. In contrast to other research on the comparison of case management languages [6, 7, 18, 32], we provide a two-fold comparison with regards to their modeling functionality and understandability:

- Functionality [method: functional comparison]: similar to related work [6, 18, 32], we use a literature analysis to deduce a set of criteria based on which the languages are compared. Additionally, we have modeled three use cases in each language to assess which of the criteria are supported.
- Understandability [method: user study]: we evaluate model understandability of the three languages with a user study, where we test the interpretation effectiveness based on task fulfilment by the users.

The results indicate that the languages have different strengths: the analytical comparison shows that fCM offers the broadest functionality out of the set of languages, while the users in the study found CMMN more understandable. This paper is based on a master’s thesis [12]. In the remainder, related work is presented in Sect. 2. Then, the method for the criteria-based comparison is given and the results presented in Sect. 3, followed by the user study with its design and its results in Sect. 4. Finally, the results are discussed and a summary and outlook is given in Sect. 5.

2 Related Work

In research, different comparisons and assessments of process modeling languages including case management languages are made. Process modeling languages

¹ <https://github.com/bptlab/chimera>, accessed 2022-03-22.

Table 1. Research works comparing case management languages

Reference	Year	Analyzed languages	Method
Pichler et al. [25]	2011	BPMN, ConDec	User Study
Di Ciccio et al. [6]	2015	YAWL, ADEPT2, SmartPM, Declare, PHILharmonic Flows, ArtiFact - GSM, MailOfMine	Criteria-based comparison
Marin et al. [18]	2015	CMMN	Criteria-based assessment
Wiemuth et al. [38]	2017	BPMN, CMMN, DMN	Use case
Zensen and Küster [39]	2018	CMMN, BPMN	Use Case
Steinau et al. [32]	2019	Custom Case Handling Approach, GSM, PHILharmonic Flows	Criteria-based comparison
Gonzalez-Lopez et al. [7]	2021	fCM, fCM Landscape	User Study
Jalali [13]	2021	CMMN and DCR	User Study

have been compared, for example, regarding their (a) capabilities to capture certain characteristics (e.g., execution depends on knowledge) [6], (b) representational capabilities and expressive power with the help of an ontology-based theory of representation (e.g., a stable state) [26] or with the help of patterns (e.g., control flow patterns) [5], (c) understandability and usability (e.g., comprehension task efficiency or perceived usefulness) [20], and practical usage [22].

Table 1 gives an overview of research works which have compared case management languages with each other or with BPMN as well as works that performed a criteria-based evaluation. The table also lists the used method, and is sorted by year to highlight the progress of such kind of research.

Pichler et al. [25] compared an imperative approach in form of BPMN models to a declarative approach in form of ConDec models by investigating the understandability with the help of a user study. This research work focuses on the comparison of the declarative to the imperative modeling paradigm.

Di Ciccio et al. [6] analyzed different case management languages to define knowledge-intensive processes and their characteristics. Based on these characteristics, different case management modeling languages are compared and evaluated. Similarly, Marin et al. [18] surveyed different definitions to find characteristics and requirements of case management. However, they focused only the extent to which CMMN fulfills the characteristics and requirements. While technically not a comparison, we include the paper here due to relevance.

Wiemuth et al. [38] intended to combine business process modeling and adaptive case management in order to model a flexible and variable medical processes. Based on the use case, CMMN and DMN are compared to BPMN. The work is limited to one use case; other case management languages are not considered. Similarly, Zensen and Küster compare modeling a use case with the flexible elements of BPMN to CMMN case model.

Steinau et al. [32] conducted an exhaustive literature survey and analyzed different data-centric process modeling approaches, including a set of case man-

agement languages. Their comparison was based on a set of criteria. As the focus was on data-centric modeling approaches, CMMN was not examined.

Gonzalez et al. [7] extended the case management language fCM for case model landscapes to ease the readability of the case models. They comparatively evaluated the landscape against the original fCM notation with a user study on model understandability. Jalali [13] investigated the perceived usefulness and ease of use of CMMN and DCR in a user study and did not find strong differences between them.

In summary, the most frequently used method for comparison in the literature is checking for criteria coverage, where the criteria were typically derived from the literature. Both existing works with user studies had a different focus, and [25] is more than ten years old and pre-dates the release of CMMN. In this work, we complement the state of the art by (i) conducting a criteria-based comparison which includes CMMN, (ii) modeling different use cases, and (iii) performing a comparative user study. In particular, our comparison includes the CMMN industry standard and two research languages under active development, fCM and PHILharmonic Flows.

3 Criteria-Based Comparison

This section presents the results of the criteria-based comparison. We assume that the readers are familiar with the modeling languages; for details of those, see [8, 15, 23], or for a concise overview see [12, Ch. 2.3]. First, we present our method in Sect. 3.1 and then, we provide the criteria and the results in Sect. 3.2.

3.1 Method

The criteria for the functional comparison were defined based on a literature study with the goal to find relevant papers on criteria-based evaluations of case management languages. We conducted the literature analysis using the knowledge databases and search engines Primo (the main knowledge database of the Technical University Berlin) and Google Scholar. The search terms were ((“Case Management” OR “Case Modeling” OR “Case Handling”) AND (“process modeling” OR “comparison” OR “analysis” OR “assessment” OR “evaluation”)). We received 18.100 results. The retrieved papers were then cleansed of medical results, as case management is still strongly linked to the healthcare domain, and reduced with regard to duplicates. Papers were included that focus on case management and requirements, a comparison, an assessment, or an evaluation. From this analysis, we identified the 14 papers referenced in Table 2 that define relevant criteria for case management.

Furthermore, requirements for the selection of criteria were defined: A criterion must be (1) universally valid, attainable, based on the characteristics of case management, (2) not redundant, and (3) relate to the case design-phase. Overall, we obtained 96 criteria, from which we removed 55 duplicates and 22 criteria relating to the case execution phase, for example the criteria “Unanticipated

exceptions” and “Flexible execution” [6]. Furthermore, we removed seven criteria for being not attainable, too general or relating to a high-level requirement, such as “Advanced collaboration” [9] or “Implicit process description” [28].

In the criteria assessment, we followed a two step approach: we first modeled use cases in each language and assessed then the fulfillment of the criteria. By modeling the use cases, a more detailed understanding of the modeling languages, their characteristics, and features is created. For criteria which were only partial or not fulfilled, we re-read the description of the modeling language to rule out biases and possible limitations from the use cases.

Three use cases were selected and textually described [12, Appendix A-C]. The use cases originate from different case management domains: medical, administrative, and consulting. The medical and the administrative process were both derived from public event logs [17,36]. Specifically, we derived textual process description by analysing the most common variants and directly-follows graphs in the process mining tool Disco. The third use case, a consultancy project planning process, was elicited by six interviews in a software engineering and consulting firm, from which we created a detailed textual process description. With the textual descriptions, we modeled the three use cases in each of the three case management languages [12, Ch. 6.1] using MS Visio. MS Visio provides the flexibility to use all notational elements present in a modeling language.

Eventually, using the insights from the modeling, the support for each criterion selected before was assessed for CMMN, fCM, and PHILharmonic Flows (PHIL) on a three-value-scale: full support, partial/implicit support, and no support. If we observed partial or no support, we checked again the description of the modeling language to validate whether a criterion was really partially or not supported.

3.2 Criteria and Analysis

Based on the literature analysis, we derived 12 relevant criteria for comparing the case management languages. The criteria, a brief description, related references, and the fulfillment of each language are shown in Table 2. The first six criteria are related to the characteristics of case management. Following are six criteria related to modeling capabilities. We discuss the criteria and results next.

Case Management Criteria. *Knowledge-driven* describes the influence of data and its availability as well as decisions made by knowledge workers on the progression of the case [6,18]. Through the case progression, the process-related knowledge evolves [19]. All three examined case management languages fully support this criterion. In CMMN, the executing knowledge worker can influence the execution of the process by making decisions based on skills and expertise. An fCM model is driven by the availability of case data expressed by conditional start events of the process fragments. Enabled fragments are executed at the discretion of the executing worker. Knowledge-driven aspects of PHIL are expressed in the micro processes, one for each case data object. Those

Table 2. Selected criteria, references and fulfillment by the three case management languages. Scale: support exceeded expected levels ‘✓✓’, full support ‘✓’, partial/implicit support ‘(✓)’, and no support ‘×’.

Name	Description	References	CMMN	fCM	PHIL
Knowledge-driven	Execution depends on the availability of data and decisions by experts	[6, 11, 18, 21]	✓	✓	✓
Data modeling	Support for the modeling of data	[4, 6, 18, 28, 32, 38]	(✓)	✓	✓
Goal modeling	Support for the modeling of goals	[4, 6, 18]	(✓)	✓	(✓)
Data-driven activities	Support for the modeling of data-driven activities	[6, 18, 28, 34]	✓	✓	✓
External events	Support for the modeling of external events	[4, 6, 7, 18]	✓	✓	×
Resource or skill modeling	Support for the modeling of resources and skills of knowledge workers	[6, 18]	(✓)	×	×
Different modeling styles	Support for different modeling styles within the modeling environment	[6, 18, 32]	(✓)	✓	✓
Management of rules and constraints	Support for the formalization and management of rules and constraints	[6, 9, 18, 25, 31, 33]	✓✓	✓	(✓)
Management of roles	Support for the modeling of roles of knowledge workers	[9, 11, 31, 38]	✓	×	✓
Management of process granularity	Support for the management of process granularity and enforcement of granularity levels	[7, 32]	(✓)	✓	✓
Specification of case data behavior	Support for the modeling the behavior of case data	[32]	×	✓	✓
Specification of interactions	Support for the modeling of interactions between processes	[4, 7, 28, 32]	(✓)	(✓✓)	×
Sum	Exceeded/full/partial/no support		1/4/6/1	0/9/1/2	0/7/2/3

data objects and its properties may then influence the execution of the macro model synchronizing the case data.

Data modeling is supported when a case management language supports the specification of a data model or its elements [18]. Explicit support for data modeling is provided when data properties and relations between data types are included [21]. CMMN allows the modeling of data in form of case file items. Nevertheless, CMMN is classified as partial support, because relations between different case data types and their properties cannot be expressed. The modeling languages fCM and PHIL both feature individual data models. Hence, both are rated as full support.

Goal modeling is supported by a case management language, if the language allows the (explicit) definition of a process goal [4]. The process goal is a global goal, and usually represents the possible termination of a case. It may be data or decision-based [6]. CMMN is rated as supporting it partially, because a separate goal definition regarding the case is not part of a CMMN model. Nonetheless, milestones and the exit criterion can be perceived as an implicit definition of a case goal. In fCM, a goal state is specified explicitly. PHIL does not require an explicit definition of a goal state, but by highlighting the final stage, goal modeling is supported implicitly.

Data-driven activities. Activities in knowledge-intensive processes depend on the related data [34]. Hence, *data-driven activities* can be represented in terms

of data conditions, but also by the influence of a separately defined data model. Data may influence the ordering, start, and end of activities [18]. CMMN, fCM and PHIL fully support data-driven activities.

External events. If external events are supported, the case modeling language allows external triggers to influence the process progression. Such a trigger originates from the process's environment and may alter data states and the sequence flow [6]. External events are included as predefined elements in CMMN. The process fragments in fCM contain external events as well. Accordingly, external events are supported completely by both languages. However, PHIL provides no support for the specification of external events.

Resources and skills Resources and skills of process-related knowledge workers are critical for case management processes [18]. The criterion examines whether it is possible to represent resources and their skills by a notational element. CMMN supports a basic or implicit support for resource and skill modeling. In fCM and PHIL, resources and skills cannot be modeled.

Criteria Regarding Modeling Capabilities. In this part, we describe the criteria related to modeling capabilities of a case management language, such as different modeling styles, the management of rules and constraints, roles, and process granularity, as well as the specification of case data behavior and the interactions.

A case consists of several elements that partially represent knowledge, e.g., data objects, separately defined skills, or behavior. Those elements might have different *degrees of structuredness* [6]. To represent the aforementioned elements appropriately, different *modeling styles* can be required, for instance, the declarative or imperative modeling style. CMMN has a strong declarative flavor [30] in defining the relations between the stages, but it also allows defining imperative parts by having a process task that links to a BPMN diagram [18]. fCM and PHIL combine both styles in their modeling notation.

In case management, *rules and constraints* are integral elements to structure a case. Thus, according to [9], case management languages are supposed to support the explicit definition of rules and constraints by the process modeler. fCM supports rules and constraints via the definition of data constraints and the usage of BPMN events (e.g., for the definition of timer constraints) in the fragments and thereby supports this criteria fully. CMMN is even more flexible, thus we recorded the support to exceed expected levels: sentries of tasks and stages allow the definition of data constraints, or rules in any rule language by defining an expression and referring the language [23] PHIL provides also the definition of data constraints. It provides a partial support because certain constraints, e.g. timer constraints, cannot be expressed.

Case management requires a *definition of roles*. The role definition has no predetermined level of precision, it can range from complex role definitions to simple roles, using e.g., only skip permissions [9]. The definition and management of roles is provided by CMMN, thus offering full support. It restricts which role is allowed to perform tasks and modify the case plan model at runtime.

However, roles have no notational element, they are only specify as attributes. fCM provides neither a role nor a permission management. Roles in PHIL can be managed and are specified as permissions in the data model. A role can generally grant reading or writing rights, marking partial support.

The degree of detail in a case management process model is described by *process granularity* [7]. Supporting this criterion are case management languages that enforce or at least recommend particular levels of granularity [32]. CMMN provides partial support for the management of process granularity by allowing the clustering of case plan items into stages. The level of granularity and thus the level of detail in fCM and PHIL is managed through the different models. In fCM, the domain model, the object lifecycle model, and the process fragments each display a different level of granularity of a case. The same applies to all components of a PHILharmonic Flows model. Hence both fCM and PHILharmonic Flows are rated to fully support the process granularity.

Specification of case data behavior means that the case management language provides support to specify the allowed behavior at runtime of the data involved in a case [32]. CMMN does not support the modeling of data object behavior. The object lifecycle model of a fCM model provides a full support and fundamentally represents a behavior model and explicitly shows how a data object behaves during process execution. PHIL depicts the behavior within its micro processes, hence the rating of full support.

Finally, the last criterion concerns whether a modeling language allows the modeling of *interactions between processes*. It requires the inclusion and visibility of the connection in the model [35]. It is irrelevant for the evaluation of the criterion whether the interacting processes are modeled in the same modeling language. Interactions between processes can be modeled in CMMN using tasks that link BPMN or other CMMN models. However, a possible data exchange between processes and the precise connections cannot be modeled. For this reason, CMMN was ranked with partial support. Also rated with partial support is fCM where interactions between processes can be depicted with message events of the BPMN language, the modeling language used for the process fragments. PHIL does not support interactions between processes.

Summary of Observations. Overall, it can be observed that fCM has the highest number of criteria fully support followed by PHIL. fCM provides language concepts for the modeling of different case management characteristics, such as data, goals, and external events. CMMN rather indirectly supports certain aspects like data, goals and resources. PHILharmonic Flows provides like fCM no modeling concept for resources, and additionally, external events cannot be captured. Less of the modeling capabilities are supported by the languages. Whereas CMMN has its strength in the management of constraints and roles, fCM and PHIL support different modeling styles, the management of process granularity and the explicit definition of case data behavior. In contrast to fCM, PHIL has the capability of role management.

4 User Study

To evaluate the model understandability of the three chosen modeling languages, we conducted an experiment in a user study. For this purpose, the users' interpretation effectiveness of different case models is compared to identify the level of model understandability. The study subjects were asked to answer questions about the case models. The main measure gathered from the experiment was the interpretation effectiveness, representing the number of correct answers [3, 16].

4.1 Hypothesis and Experiment Design

In this experiment, we follow the guidelines for empirical evaluations of modeling languages, proposed by Burton-Jones et al. [3]. When planning the study, we also considered the guidelines for experimental design by Juristo and Moreno [14].

To compare the case management languages used for this experiment, we defined response variables. The dependent variables are *effectiveness* and *perceived difficulty*. The effectiveness is measured by the number of correct answers and may range between 0 and 15, as 15 questions per process model are to be answered. The perceived difficulty is rated by the participants directly and may range from 1 (very easy) to 5 (very hard). The independent variable is the modeling language. It is predefined and not modifiable by study subjects. As mentioned, CMMN is an industry standard and has undergone an exhaustive development process. Thus, we hypothesized that, comparison to the other two languages, CMMN performs higher in terms of measured effectiveness, and lower in perceived difficulty of CMMN. As such, we defined the following hypotheses and corresponding null hypotheses by using the dependent variables effectiveness and perceived difficulty:

- $H1_A$: The measured effectiveness of CMMN is higher than the measured effectiveness of fCM and PHIL.
- $H1_0$: There is no significant difference in the measured effectiveness of CMMN, fCM and PHIL.
- $H2_A$: The perceived difficulty of the CMMN is lower than the perceived difficulty of fCM and PHIL.
- $H2_0$: There is no significant difference in the perceived difficulty of CMMN, fCM and PHIL.

The experiment follows a crossover design [7]. Each subject receives three case models, each representing one of the previously modeled use cases [12, Ch. 6.1] (i.e., sepsis treatment, purchase handling and consultancy project planning) and one of the modeling languages (CMMN, fCM and PHILharmonic Flows (PHIL)). We aimed to mitigate possible object learning effects by presenting each process only exactly once to each subject, and technique learning effects by using each modeling language only exactly once per questionnaire. Furthermore, we changed the sequence in which the modeling languages are presented to the participants to minimize effects of tiredness. This results in the six possible combinations illustrated in Table 3. All case models were checked to be of

Table 3. Combinations

#	Sepsis process	Purchase handling process	Procurement process
1	CMMN	fCM	PHIL
2	CMMN	PHIL	fCM
3	fCM	CMMN	PHIL
4	fCM	PHIL	CMMN
5	PHIL	fCM	CMMN
6	PHIL	CMMN	fCM

comparable complexity by aligning the number of activities, events, and particularities, if necessary. Semantic equivalence was ensured by checking in the group of co-authors that each understanding question results in the same answer for all three languages. Few exceptions were in aspects that can not be expressed by the language, e.g. relation between case data in CMMN, roles in fCM, timer constrains in PHIL. In those cases, the indented right answer for the question is “I don’t know”. During the experiment, the models were constantly available, as proposed by Parson and Cole [24].

4.2 Experiment Implementation

The subjects were BSc and MSc students, PhD candidates and professionals who were invited to voluntarily participate in the anonymous experiment. The students were from the University of Potsdam and the Technische Universitaet Berlin. The students participating can be considered future users of business process management, including case management. The entirety of subjects in this experiment represents the target audience of case management as they have a basic knowledge of business process modeling and/or work in technical fields where process modeling is applied.

The questions, the provided material, and the models were solely available in English. Each subject was randomly assigned one of the combinations from Table 3, but with consideration of an equal distribution among the combinations.

The experiment was conducted online using Google Forms. The questionnaire was available for 2.5 weeks in May 2021. Due to COVID-19 restrictions, we were not able to conduct the experiment under laboratory conditions. All invited subjects received a link to the Google Forms questionnaire. The answers were automatically logged and checked for correctness by the platform. Only fully completed questionnaires were considered. We received 26 complete responses, distributed over the six combinations. Participants reported that they needed between 30 and 60 min for partaking in the study. The questionnaire and material was structured into four parts (1) demographic questions;(2) process modeling questions used to assess the participant’s experience in business process modeling and case management; (3) a brief overview and introduction to the used case management modeling languages in the form of specially produced videos;

Table 4. Number of User Types per combination with User Type A (no or basic knowledge in process modeling), User Type B (advanced knowledge in business process modeling) and User Type C (professional knowledge in business process modeling or prior knowledge in Case Management approach)

#	User type A	User type B	User type C	Σ
1	1	1	2	4
2	1	1	1	4
3	1	2	2	5
4	2	1	1	4
5	3	1	1	5
6	1	3	0	4
Σ	9	10	7	26

(4) understandability test consisting of 15 questions for each presented process model, and user feedback in form of rating the perceived difficulty and an open feedback question.

The questions in the fourth part were arranged in random order, and participants selected an answer out of the options “true”/“false”/“I don’t know”. The “I don’t know” option enabled participants to mark that an answer could not be given on the basis of the provided model, or to avoid guessing in case of uncertainty. The questions were formulated consistently with regard to case-related characteristics to ensure comparable difficulty. The effectiveness was then measured as the total score for each model separately. Out of the three options, exactly one was correct in each case, i.e., in some cases the “I don’t know” option was rated as correct if an aspect was not present in the corresponding model. A correct answer translates to one point, which means 15 points could be achieved per model. For each of the three models, participants rated the perceived overall difficulty and had an option to provide free-text feedback on the case model. The full questionnaire including all questions can be found in [12, pp. 73].

4.3 Experiment Results

After the experiment was concluded, the validity of the data was analyzed. The analysis of plausibility and consistency revealed that all responses are valid and were then used in the subsequent statistical analysis. All results presented in this section, and the related summaries and conclusions, relate solely to this experiment and do not claim general validity.

26 subjects participated with 15 students and 11 postgraduates. In Table 4, the number of participants per combination is shown, categorized by their knowledge on process modeling. In summary, we achieved a good distribution of participants with different knowledge regarding process modeling over the different questionnaire combinations. Only two subjects stated to have no knowledge in business process modeling, all other participants had at least basic knowledge. Also, only one of the participants had already worked with case management, all other users of type C had professional knowledge in BPMN. The data shows

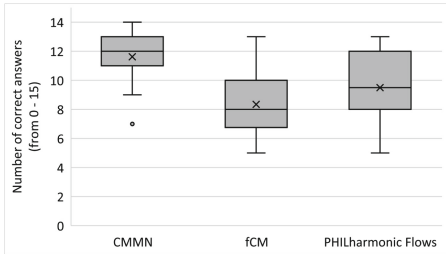


Fig. 1. Effectiveness (average marked with “X” and median with “-”).

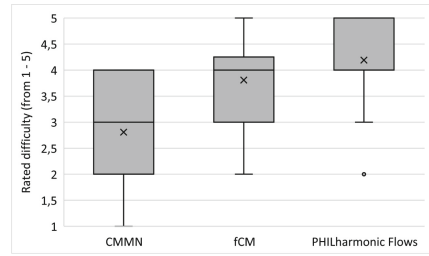


Fig. 2. Perceived Difficulty (average marked with “X” and median with “-”).

Table 5. Hypothesis testing results

Sub-hypothesis	Sig. (p)	Status	Hypothesis	Status
$H1_0^{CMMN-fCM}$	0.00	rejected		
$H1_0^{CMMN-PHIL}$	0.00	rejected	$H1_0$	rejected
$H1_0^{fCM-PHIL}$	0.03	rejected		
$H2_0^{CMMN-fCM}$	0.00	rejected		
$H2_0^{CMMN-PHIL}$	0.00	rejected	$H2_0$	rejected
$H2_0^{fCM-PHIL}$	0.08	not rej		

that, among our participants, BPMN is by far the most used and known process modeling language of the four considered, while CMMN is ahead of fCM and PHIL.

Figure 1 and 2 provide the resulting effectiveness and perceived difficulty for each of the case management modeling languages. The effectiveness is depicted in form of a score that can range from 0 to 15. It represents the number of correctly answered questions. The average score for CMMN is 11.62 with a median of 12. For fCM, the average score was 8.35 and the median 8. The average and median scores of PHIL are identical and equal 9.5². According to the analysis of the measured effectiveness in this experiment, CMMN is the most understandable modeling language, followed by PHIL and lastly fCM. However, PHIL has a higher scattering of the effectiveness data. Figure 2 implies that PHIL is perceived as the most difficult case modeling language, while CMMN appears to be the easiest of the modeling languages. Nevertheless, the median of the perceived difficulty of fCM and PHIL is equal at 4.

Following the analysis, the hypotheses introduced above were tested. In this experiment, the statistical analysis considered a 95% confidence. The data used for the hypothesis testing consists of paired measurements. For testing, the hypotheses were subdivided for pairwise comparison of languages. For instance, $H1_0^{CMMN-fCM}$ is the sub-hypothesis that states that no difference exists between

² Median is not an integer, which happened since the number of data points is even.

CMMN and fCM in terms of effectiveness. The paired measurements are the scores or perceived difficulty ratings for one process modeled in two modeling languages. The subjects are equivalent to the participants, which provides the independence of subjects. As the measured differences of the data are non-normally distributed, a non-parametric alternative to the t-test is applied: the Wilcoxon signed rank test.

Table 5 showing the results of the pair-wise testing demonstrates that the hypothesis testing provided significant evidence for differences in effectiveness between CMMN, fCM and PHIL. From analyzing the data and intermediate results from the Wilcoxon signed rank test, we derived that CMMN has a higher understandability than fCM ($T = (-)0$) and PHIL ($T = (-)13$). The analysis further indicates that PHIL has a higher effectiveness and therefore a better understandability than fCM ($T = (+)67,5$). From the results for the sub-hypotheses, we can deduce that H_{10} can be rejected as well. Hypothesis testing provides significant statistical evidence to reject the assumption that CMMN is perceived as difficult as fCM and PHIL. Conversely, by further analysis we found that CMMN is perceived as easier than fCM ($T = (+)20$) and PHIL ($T = (+)5,5$) by the participants. Finally, hypothesis $H_{20}^{\text{fCM-PHIL}}$ cannot be rejected by hypothesis testing, indicating that no significant difference in the perceived difficulty of the two exists. However, this does not factor into H_{20} . Given the results for sub-hypotheses $H_{20}^{\text{CMMN-fCM}}$ and $H_{20}^{\text{CMMN-PHIL}}$, we also reject H_{20} .

From the comments given for the different models, we were able to gather feedback for a brief qualitative analysis. The comments on CMMN state that the modeling language is “more readable in comparison” and questions were considered “easier to answer”. Nevertheless, one participant commented that the model was not very informative due to only one type of connector, a lack of phasing, and a general lack of time dependencies. Secondly, the comments on fCM focus on the complexity and associated difficulty of readability. Also, obtaining an overview of the case model is considered difficult. However, it was noted that the process fragments were easy to understand due to the BPMN notation. Finally, PHIL was repeatedly called the hardest of the three in the comments. To one participant the connection of the different micro processes and macro process of PHIL stayed unclear.

5 Discussion and Outlook

Main Observations. Based on the two-fold comparison, the main observations for the different case modeling languages are the following. *fCM* performs strongest out of the three languages in the analytical comparison with the highest full coverage of the selected case management and modeling capability criteria. However, it has the lowest interpretation effectiveness in the comparison and performs similar to PHILharmonic Flows (PHIL) regarding the perceived interpretation difficulty. Therefore, Gonzalez et al. [7] have developed a case landscape for fCM to improve the model interpretation effectiveness of users. In contrast, *CMMN* supports less of the functional criteria fully, but has both a

higher interpretation effectiveness and a lower perceived difficulty. Based on the insight from literature [2], this could be explained with less modeling concepts of CMMN in contrast to fCM and PHIL, which could improve the interpretation effectiveness of people. The data-oriented case management approach *PHIL* supports fewer of the case management criteria, but supports more criteria as fCM regarding the modeling capabilities. Subjects perceived PHIL as more difficult, but the interpretation effectiveness of the language was higher than fCM. A possible reason could be that PHIL focuses on the case data and their life cycles, whereas fCM uses multiple model types representing the process fragments on the one hand, and the involved data on the other hand.

Threats to Validity. In the analytical comparison, criteria were selected based on a structured literature search. Still, they have not been checked for completeness and practical relevance, which could be done in future. When assessing the criteria, we took certain measures to reduce the bias. First, case models based on real-world scenarios were modelled. When certain criteria were not or only partially supported, we checked again the respective description of the modeling language to rule out limitations from the use cases.

Regarding the experiments: domain knowledge of participants could influence the results which should be investigated in future works. Alternatively, the possibility of domain knowledge could be eliminated by providing business process models labeled with abstract symbols, like letters, instead of task descriptions. Furthermore, comparable difficulty of statements could be ensured more precisely by not only using guidelines for the formulation but by pre-tests. Comparable process model complexity could have been determined by different methods than the model metrics used here.

Concerning the implementation, it was not possible due to COVID-19 restrictions to conduct the experiment under laboratory conditions. In a laboratory experiment, the proposed variables' interpretation effort and efficiency could be included, as a meaningful measurement of time would be possible [3]. The number of participants is sufficient to perform significance analyses. Still, the significance of the results would likely be higher with more subjects participating in the experiment. The influence of prior knowledge of business process modeling, familiarity with case management, and the individual modeling languages on the results should be examined and evaluated in further analysis through statistical analyses. Overall, all results from this study should be validated in further experiments, ideally taking the described threats to validity into consideration.

Summary and Outlook. In this work, we conducted a structured comparison of the case management languages CMMN, fragment-based Case Management (fCM), and PHILharmonic Flows (PHIL) with regards to their modeling functionality and applicability. This comparison extends existing research in that it not only analyzes functional aspects but also the understandability. In the context of our study, the results show that CMMN provides better comprehensible

case models in contrast to the other two languages, but it provides less functional support. Also, there is no broad usage in case engines yet [29]. It might be useful as an intermediate for business users and could be translated into other case management languages, such as fCM or PHIL; to generate models which could then be used to verify or execute the cases. The method we established in this research work could also be used as a framework to compare the modeling capability of case management languages. Currently, it focuses on CMMN, fCM, and PHIL. In future, it could be used to evaluate further existing languages, such as DCR graphs. The modeling capabilities have been evaluated regarding their coverage of requirements of knowledge-intensive processes, taken from the literature. Still, in the future pattern-based or ontological-based comparison regarding the representational capabilities [27] could be conducted. Furthermore, the focus of this comparison is on modeling and can be extended to execution, monitoring and analysis capabilities of case management languages in the future.

References

1. Andrews, K., Steinau, S., Reichert, M.: Enabling runtime flexibility in data-centric and data-driven process execution engines (2021). IS 101, 101447
2. Bajaj, A.: The effect of the number of concepts on the readability of schemas: an empirical study with data models. *Requirements Eng.* **9**(4), 261–270 (2004)
3. Burton-Jones, A., Wand, Y., Weber, R.: Guidelines for empirical evaluations of conceptual modeling grammars. *J AIS* **10**(6), 495–532 (2009)
4. de Man, H.: Case management: a review of modeling approaches. *BPTrends*, January (2009)
5. van Der Aalst, W.M., Ter Hofstede, A.H., Kiepuszewski, B., Barros, A.P.: Workflow patterns. *Distrib. Parallel Databases* **14**(1), 5–51 (2003)
6. Di Ciccio, C., Marrella, A., Russo, A.: Knowledge-intensive processes: characteristics, requirements and analysis of contemporary approaches. *JoDS* **4**(1), 29–57 (2015). <https://doi.org/10.1007/s13740-014-0038-4>
7. Gonzalez-Lopez, F., Pufahl, L., Munoz-Gama, J., Herskovic, V., Sepúlveda, M.: Case model landscapes: toward an improved representation of knowledge-intensive processes using the fCM-language. *SoSym* **20**, 1–25 (2021)
8. Haarmann, S., Holfter, A., Pufahl, L., Weske, M.: Formal framework for checking compliance of data-driven case management. *JoDS* **10**, 1–21 (2021)
9. Hauder, M., Pigat, S., Matthes, F.: Research challenges in adaptive case management: a literature review. In: 2014 IEEE 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations (EDOCW) (2014)
10. Hewelt, M., Weske, M.: A hybrid approach for flexible case modeling and execution. In: La Rosa, M., Loos, P., Pastor, O. (eds.) *BPM 2016*. LNBIP, vol. 260, pp. 38–54. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45468-9_3
11. Hinkelmann, K., Pierfranceschi, A.: Combining process modelling and case modelling. In: *MeTTeG14*, pp. 83–93. Universitas Studiorum, Mantova (2014)
12. Holz, J.: A Systematic Comparison of Case Management Approaches. Master's Thesis, TU Berlin (2021). <https://doi.org/10.13140/RG.2.2.22243.68649>
13. Jalali, A.: Evaluating perceived usefulness and ease of use of CMMN and DCR. In: Augusto, A., Gill, A., Nurcan, S., Reinhartz-Berger, I., Schmidt, R., Zdravkovic, J. (eds.) *BPMDS/EMMSAD -2021*. LNBIP, vol. 421, pp. 147–162. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-79186-5_10

14. Juristo, N., Moreno, A.M.: *Basics of Software Engineering Experimentation*. Springer, Boston (2001)
15. Künzle, V., Reichert, M.: PHILharmonicFlows: towards a framework for object-aware process management. *J. Softw. Evol. Process.* **23**(4), 205–244 (2011)
16. Malinova, M.: *A Language for Designing Process Maps*. Ph.D. thesis, WU Vienna (2016)
17. Mannhardt, F.: Sepsis cases - event log. <https://doi.org/10.4121/UUID:915D2BFB-7E84-49AD-A286-DC35F063A460>
18. Marin, M.A., Hauder, M., Matthes, F.: Case management: an evaluation of existing approaches for knowledge-intensive processes. In: Reichert, M., Reijers, H.A. (eds.) *BPM 2015. LNBP*, vol. 256, pp. 5–16. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42887-1_1
19. Marjanovic, O., Freeze, R.: Knowledge intensive business processes: theoretical foundations and research challenges. In: 44th HICSS. IEEE, Piscataway (2011)
20. Mendling, J., Strembeck, M., Recker, J.: Factors of process model comprehension-findings from a series of experiments. *DSS* **53**(1), 195–206 (2012)
21. Motahari-Nezhad, H.R., Swenson, K.D.: Adaptive case management: overview and research challenges. In: 2013 IEEE 15th Conference on Business Informatics (2013)
22. Muehlen, Michael zur, Recker, Jan: How much language is enough? Theoretical and practical use of the business process modeling notation. In: Bubenko, J., Krogstie, J., Pastor, O., Pernici, B., Rolland, C., Solvberg, A. (eds.) *Seminal Contributions to Information Systems Engineering*, pp. 429–443. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36926-1_35
23. Object management group: case management model and notation (CMMN): specification. version 1.1 (2016)
24. Parsons, J., Cole, L.: What do the pictures mean? Guidelines for experimental evaluation of representation fidelity in diagrammatical conceptual modeling techniques. *Data Knowl. Eng.* **55**(3), 327–342 (2005)
25. Pichler, P., Weber, B., Zugal, S., Pinggera, J., Mendling, J., Reijers, H.A.: Imperative versus declarative process modeling languages: an empirical investigation. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) *BPM 2011. LNBP*, vol. 99, pp. 383–394. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28108-2_37
26. Recker, J., Rosemann, M., Indulska, M., Green, P.: Business process modeling—a comparative analysis. *J AIS* **10**(4), 1 (2009)
27. Recker, J., Rosemann, M., Krogstie, J.: Ontology-versus pattern-based evaluation of process modeling languages: a comparison. *Commun. Assoc. Inf. Syst.* **20**(1), 48 (2007)
28. Reijers, H.A., Rigter, J.H.M., van der Aalst, W.M.P.: The case handling case. *Int. J. Coop. Inf. Syst.* **12**(03), 365–391 (2003)
29. Routis, I., Bardaki, C., Dede, G., Nikolaidou, M., Kamalakis, T., Anagnostopoulos, D.: CMMN evaluation: the modelers’ perceptions of the main notation elements. *SoSym* **20**, 1–21 (2021). <https://doi.org/10.1007/s10270-021-00880-3>
30. Slaats, T.: Declarative and hybrid process discovery: recent advances and open challenges. *JoDS* **9**(1), 3–20 (2020). <https://doi.org/10.1007/s13740-020-00112-9>
31. Slaats, T., Mukkamala, R.R., Hildebrandt, T., Marquard, M.: Exformatics declarative case management workflows as DCR graphs. In: Daniel, F., Wang, J., Weber, B. (eds.) *BPM 2013. LNCS*, vol. 8094, pp. 339–354. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40176-3_28

32. Steinau, S., Marrella, A., Andrews, K., Leotta, F., Mecella, M., Reichert, M.: DALEC: a framework for the systematic evaluation of data-centric approaches to process management software. *SoSym* **18**(4), 2679–2716 (2019). <https://doi.org/10.1007/s10270-018-0695-0>
33. van der Aalst, W.M.P., Pesic, M., Schonenberg, H.: Declarative workflows: balancing between flexibility and support. *CSRD* **23**(2), 99–113 (2009). <https://doi.org/10.1007/s00450-009-0057-9>
34. van der Aalst, W.M., Weske, M., Grünbauer, D.: Case handling: a new paradigm for business process support. *Data Knowl. Eng.* **53**(2), 129–162 (2005)
35. van der Aalst, W.M., Weske, M., Wirtz, G.: Advanced topics in workflow management: issues, requirements, and solutions. *JIDPS* **7**(3), 49–77 (2003)
36. van Dongen, B.F.: BPI challenge (2019). <https://doi.org/10.4121/uuid:d06aff4b-79f0-45e6-8ec8-e19730c248f1>
37. White, M.: Case mgmt.: combining knowledge with process. *BPTrends* (2009)
38. Wiemuth, M., Junger, D., Leitritz, M.A., Neumann, J., Neumuth, T., Burgert, O.: Application fields for the new object management group (OMG) standards case management model and notation (CMMN) and decision management notation (DMN) in the perioperative field. *IJCARS* **12**(8), 1439–1449 (2017). <https://doi.org/10.1007/s11548-017-1608-3>
39. Zensen, A., Küster, J.: A comparison of flexible BPMN and CMMN in practice: a case study on component release processes. In: *EDOC*, pp. 105–114. *IEEE* (2018)