Computer Science and Information Systems 19(1):455–493 https://doi.org/10.2298/CSIS210423066B

Automatic Derivation of the Initial Conceptual Database Model from a Set of Business Process Models*

Drazen Brdjanin[1], Aleksandar Vukotic[2], Danijela Banjac[1],
Goran Banjac[1], and Slavko Maric[1]

[1] University of Banja Luka, Faculty of Electrical Engineering, Patre 5
78000 Banja Luka, Bosnia and Herzegovina
{drazen.brdjanin,danijela.banjac,goran.banjac,slavko.maric}@etf.unibl.org
[2] Automovens Ltd, Jovana Ducica 23a
78000 Banja Luka, Bosnia and Herzegovina
aleksandar.vukotic@rt-rk.com

Abstract. The article presents an approach aimed at automatically deriving the initial conceptual database model from a set of business process models. The approach proposes the incremental synthesis of the target model by iteratively composing the partial conceptual database models that are derived from the models contained in the source set. The approach is implemented by the AMADEOS tool, which is the first online web-based tool enabling the automatic derivation of the conceptual database model from a set of business process models. The experimental evaluation proves that the implemented approach enables effective automatic derivation of the initial conceptual database model.

Keywords: AMADEOS, BPMN, Business Process Model, Class Diagram, Conceptual Database Model, Model Composition, UML.

1. Introduction

With the development of the model-driven paradigm, *business process models* (BPMs) are playing an increasingly important role in the field of information systems and software engineering, serving as a basis for generation of the target system models. In our research we focus on using BPMs as a basis for the *model-driven synthesis of data models* (MDSDM). Furthermore, some recent experiments [7,20] imply that well-formed data-centric BPMs enable effective and efficient automatic synthesis of *conceptual database models* (CDMs).

Motivation. Although the idea of MDSDM, taking BPMs as a starting base, dates back in the early 1990s, in the existing literature there is only a small number of papers presenting the implemented automatic model-driven generator of the target data model with the corresponding evaluation results, while the great majority of papers only present modest achievements in (semi)automated, or even manual, data model synthesis. The surveys [17,69,14] show that the existing approaches are characterized by the *direct*

* This article constitutes an extended version of the conference paper entitled "*Automatic Derivation of Conceptual Database Model from a Set of Business Process Models*" presented at *INISTA – 2020 (International Conference on Innovations in Intelligent SysTems and Applications)*, August 24-26, 2020, Novi Sad, Serbia.

synthesis of the target model based on BPMs represented by a single concrete notation such as BPMN [58] or UML [59] activity diagram (UML AD). Furthermore, the existing tools are mainly implemented as plug-ins or transformation programs within development platforms (typically Eclipse-based), and also able to process only a single BPM (represented by a single diagram), although a real source model contains a finite set of models (diagrams). A more detailed overview of the related work is provided later in the article.

The surveys [17,69,14] show that a fully automated MDSDM approach is still the subject of extensive research, and the aforementioned problems of limited functionality, portability, effectiveness and efficiency of the existing MDSDM tools remain a significant research goal. In order to contribute the solving these problems, we have started a long-term research project focused on the automatic CDM synthesis based on a collection of models representing business processes of an enterprise, with the main objective to develop an online platform-independent tool for the automatic CDM derivation from a collection of BPMs that may be represented by different notations, and differently serialized, as well. The first very important milestone in our progress was the indirect two-phase CDM synthesis [21], which enables and facilitates derivation of the target data model from differently represented BPMs. By applying the two-phase CDM synthesis approach, we have launched the AMADEOS[3] system [19,36,14], which was the first online web-based system enabling automatic CDM derivation from BPMs that may be represented by two different notations and differently serialized. However, AMADEOS was able to automatically generate a CDM based on a single BPM, as the large majority of the existing MDSDM tools. A more detailed overview of the project progress is provided later in the article.

Objectives. The last-mentioned limitation of the AMADEOS system, i.e. its ability to automatically derive an initial CDM only from a single BPM, directly motivated our research presented in this particular article, with the following main objectives:

(1) *find an appropriate way to achieve automatic CDM derivation from a collection of BPMs, and extend the existing functionality of the AMADEOS system by applying the given aproach;*
(2) *evaluate to what extent is the improved AMADEOS system able to automatically generate the target initial CDM by applying this approach.*

Contributions. In order to achieve the first research objective, we propose the incremental synthesis of the target model by iteratively merging the partial CDMs that are derived from the BPMs contained in the source set. Such an approach enables the automatic synthesis of the target CDM by retaining and exploiting all existing capabilities of MDSDM tools (AMADEOS, in this case). This constitutes the first main contribution of this particular research. By applying this approach, we obtained an online web-based tool that publicly provides the MDSDM functionality by enabling the automatic CDM synthesis based on the set of BPMs represented by two concrete notations: BPMN and UML AD. This constitutes the second main contribution of this research. In order to achieve the second research objective, we performed very extensive case study-based and experimental evaluation focused on the approach effectiveness, through the assessment of correctness

[3] Available at: *http://m-lab.etf.unibl.org:8080/amadeos/*

and completeness of the CDMs automatically derived from the sets of BPMs. The experimental results prove that the implemented approach enables effective automatic derivation of the initial CDM (particularly for classes).

Some of these research results have been already presented at the *INISTA-2020 Conference*. This particular article constitutes an extended version of the corresponding conference paper [26], which is extended by: (i) a detailed overview of the related work, (ii) a detailed presentation of the proposed approach, and (iii) the results of the experimental evaluation of the implemented approach.

Article organization. The article is structured as follows. After this introductory section, Section 2 presents the related work. Section 3 presents the approach, while Section 4 presents the implemented tool. Section 5 provides an illustrative example of automatic CDM derivation from a set of BPMN models. Section 6 presents the evaluation of the implemented approach. Finally, Section 7 concludes the article and gives directions for future work.

2. Related Work

In this section we firstly provide an overview of the existing MDSDM approaches[4], then we position our approach and present the advancements in comparison with our previous work and other related approaches.

Overview of the existing approaches. The existing MDSDM approaches, regarding the primary focus of the source notation, can be classified into four main categories [17]: *process-oriented*, *goal-oriented*, *function-oriented*, and *communication-oriented*.

Figure 1 presents a chronological overview of the existing MDSDM approaches, grouped by the source notation. Different marks are used to differentiate: (i) *completeness of the source model* – a source model can be *complete* or *partial* (partial model contains a single diagram, although a real model contains a finite set of diagrams), and (ii) *automatization level of the approach*, which can be *manual* (not supported by any software tool), *semiautomatic* (supported by a tool, but designer's assistance is still required), or *automatic* (without designer's assistance). The arrows depict improvements in the same approach, presented in different papers.

POM-based approaches. Our approach belongs to the most dominant category of the MDSDM approaches that take *process-oriented models* (POMs) as a basis for the MDSDM. Although the first POM-based approach [75] was proposed back in 1990, the boom of the POM-based MDSDM approaches was highly influenced in the last 15 years by the development of metamodel-based notations (particularly UML AD and BPMN) and specialized model-to-model transformation languages (ATL[5] and QVT[6]). Apart from BPMN and UML AD, which are dominantly used, the existing POM-based approaches also take source models represented by: Petri Net, RAD (Role Activity Diagram), EPC (Event-driven Process Chain), TCD (Task Communication Diagram), and A-graph.

[4] For a more detailed survey we refer the readers to [17].

[5] ATLAS Transformation Language [44]

[6] Query/View/Transformation [57]

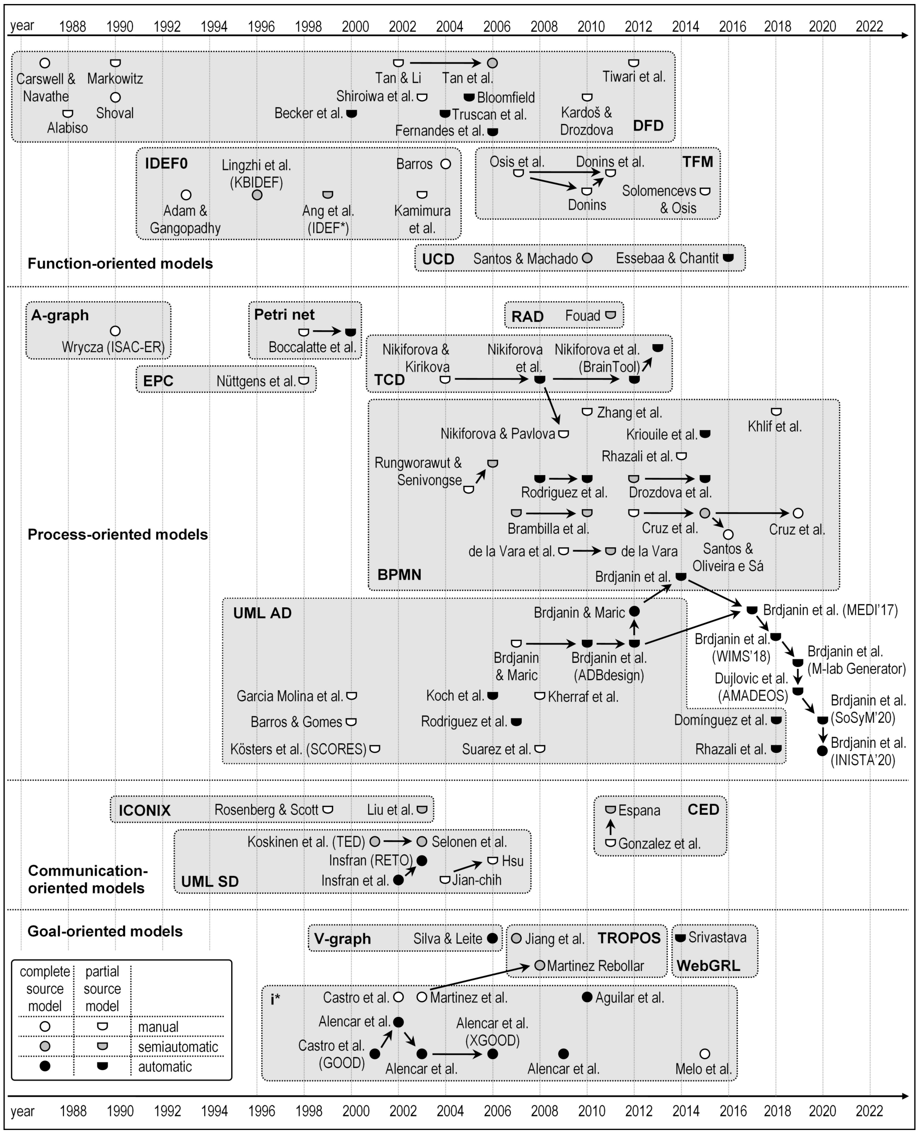


Fig. 1. Chronological overview of MDSDM approaches (based on [17,14])

Although more than 40 POM-based papers have been published in the last 30 years, there is still no approach and the corresponding tool that enable automatic synthesis of the complete target model, whereby the most approaches allow for (semi)automated synthesis with modest correctness and completeness. Moreover, many of them are based on guidelines and informal rules that do not allow automatic synthesis at all. Finally, almost all approaches consider a single process model as a basis for MDSDM, although a real model constitutes a collection of process models. Only three papers [75,25,29] consider the complete source model, but only [25] presents the implemented tool (*ADBdesign*).

Among about twenty papers that consider the BPMN-based MDSDM, only a few papers [64,63,49,34] present some tool (mainly ATL- and QVT-based transformation programs) enabling the automatic MDSDM, but with very low effectiveness. The semi-automatic BPMN-based MDSDM is presented in [65,12,13,32,35], while the other proposals [76,56,35,45,31] are not implemented at all. Regarding the formalism level of the existing BPMN-based approaches, the formal rules are presented in [23,66], and partially in [30,29,31], while the others give only the informal guidelines. Regarding the source model completeness, only three papers [29,31,66] consider a collection of the source models, but none of the proposed approaches have been implemented.

Among more than ten papers that consider UML AD as a basis for MDSDM, only [25] considers a collection of the source models and presents the implemented automatic tool (*ADBdesign*), while the majority [46,47,62,18,15,24,16,33,61] present the automatic data model derivation from incomplete source models, but with very low effectiveness.

There are also several related papers proposing the usage of TCD notation as a starting point for MDSDM, initially through an intermediate model, while [54] presents the BrainTool generator, which generates the data model directly from TCD. However, like the majority, they do not consider the complete source model. Among the other POM-based approaches, only two papers [11,40] present tools for the (semi)automatic data model synthesis based on the partial source model.

Other MDSDM approaches. Since our approach belongs to the POM-based approaches, here we provide only a short overview of other MDSDM approaches.[7]

The *function-oriented models* (FOMs), used as a basis for MDSDM, are represented by four different notations: DFD (Data Flow Diagram), IDEF0, TFM (Topological Functioning Model), and UML UCD (Use Case Diagram). Although the first ideas about the FOM-based MDSDM appeared back in the late 1980s, the survey shows that the semantic capacity of FOMs has not been sufficiently identified to enable automatic synthesis of the complete target data model. The large majority of the approaches are based on guidelines and informal rules, and take an incomplete source model as the basis. The automatic data model generation is presented in [9,73,39,10,38], while the semi-automatic generation is presented in [72,51,6,67].

The *goal-oriented models* (GOMs), used as a basis for MDSDM, are represented by the i* notation and some i*-originated notations like TROPOS, V-graph, and WebGRL. The automatic (to some extent) GOM-based MDSDM, based on the complete source model, is presented in [27,4,3,5,70,43,53,2,1,71].

The *communication-oriented models* (COMs), used as a basis for MDSDM, are represented by three different notations: ICONIX (Robustness Diagram), CED (Communicative Event Diagram) and UML SD (Sequence Diagram). The automatic data model synthesis, based on the complete source model, is presented in [41,42], while the semi-automatic synthesis is presented in [48,68,52,37].

Evaluation of the existing approaches. The large majority of all existing MDSDM approaches are not evaluated at all. Most of the papers reporting evaluation results mainly focus on approach usability, but not on the qualitative and/or quantitative assessment of the implemented tools or generated data models.

The GOM-based approaches are not evaluated.

[7] For more detailed overviews, we refer the readers to [17,20].

Only one COM-based approach [37] is evaluated based on lab demos and an experiment with students (reported model completeness is $\sim$70%).

Only [72] presents evaluation results of a FOM-based approach, but the authors do not focus on the assessment of the method effectiveness and efficiency.

Regarding the POM-based MDSDM approaches, the case-study based evaluation results are reported in [23,25,55], while the results of controlled experiments are reported in [32,40,7,22,20]. The most complete evaluation results, which are based on the experiment conducted with a significant number of practitioners, are presented in [22,20] (average completeness of the generated models is over 80%).

Comparison with our previous work and other related approaches. In this article we present the most recent developments in a long-term research project that is aimed at automatic BPM-driven CDM synthesis. As depicted in Fig. 1, the initial ideas for the manual data model derivation were presented more than ten years ago, while the first implementation (*ADBdesign*) was presented back in 2010 [18]. After the transformation rules were upgraded and formalized [16], the automatic MDSDM from a collection of the UML ADs was presented in 2012 [25]. Based on the experimentally confirmed [7,20] semantic capacity of data-centric BPMs, a two-phase BPM-driven approach to the CDM synthesis was proposed [21], and the first online service-oriented BPM-driven CDM generator (*M-lab Generator*) was implemented [19]. Finally, the AMADEOS system [36,14] was launched, as the first online web-based system for BPM-driven CDM synthesis. AMADEOS implements the CDM synthesis process through an orchestration of the *M-lab Generator* services, and enables the CDM synthesis based on BPMs that may be represented by BPMN or UML AD. Since the given approach enables the CDM synthesis based on differently represented source BPMs, all the developments after [21] are depicted outside of any POM region in Fig. 1.

The pre-existing AMADEOS release [14] was able to automatically derive the target model from a single BPM. In this article we present the most recent achievement in the entire project and development of the AMADEOS system – expanding its functionality to enable the automatic CDM derivation from the set of BPMs. In this way, we obtained the first online tool that publicly provides the MDSDM functionality based on a set of BPMs that may be represented by two different notations (BPMN or UML AD) and differently serialized (XMI[8] or XSD[9]).

Unlike the other existing MDSDM tools, AMADEOS is not dependent on any particular modeling platform and enables automatic CDM generation in a web browser, without any installation of tools or plug-ins. It enables users to upload a collection of BPMs, to generate the CDM, as well as to export the generated CDM in the XMI format for further processing in some other database design tool. In this way, AMADEOS may be very beneficial both for industrial and academic purposes – database designers are provided with a tool for BPM-based MDSDM, software engineers are provided with online services that may be invoked from their own tools, while researchers are able to compare their tools against AMADEOS. Currently, there are no other publicly available online tools that enable CDM generation based on a collection of BPMs, only some indications of plug-ins and transformation programs that are not publicly available.

[8] XML Metadata Interchange

[9] XML Schema Definition

3. Approach to Automatic CDM Derivation

This section presents the proposed approach to automatic CDM derivation from a set of BPMs. Firstly, we provide an overview of the whole process and the corresponding high-level algorithm for the incremental CDM synthesis. Then we present the major functional building blocks of the entire approach: (1) semantic capacity of the BPMs for the automatic CDM synthesis, (2) two-phase synthesis of the partial CDMs, and (3) composition of the partial CDMs. Finally, we present applied techniques to overcome some inconsistencies in the source set of BPMs.

3.1. Incremental CDM Synthesis

The starting point for the automatic CDM derivation is a set BM that contains a finite number of BPMs, i.e. $BM = \{bpm_1, \ldots, bpm_i, \ldots, bpm_n\}$.

The approach proposes the incremental synthesis of the target CDM (denoted by cdm) by iteratively composing the partial CDMs that are derived from the models contained in the source set BM. The entire process is formally specified by the high-level algorithm presented in Fig. 2, and illustrated in Fig. 3. The target model cdm is initially empty. For each model bpm_i from the source set BM, we obtain the corresponding partial CDM (cdm_i) by applying the appropriate transformation $\mathcal{G}$. This partial CDM and the result (cdm) of all previous iterations are composed ($\oplus$ operator) into the resulting model cdm. After processing all models from the source set BM, the resulting model represents the target CDM. The presented approach to the incremental synthesis of the target CDM corresponds to the well-known *binary ladder integration strategy* [8] of partial (conceptual) schemas.

The aforementioned high-level transformation $\mathcal{G}$ represents the entire process of the CDM derivation from a single source BPM. It is described in the following two subsections (3.2 and 3.3), while subsection 3.4 presents the model composition.

```
$cdm \leftarrow \emptyset$
for all $bpm_i \in BM$ do
    $cdm_i \leftarrow \mathcal{G}(bpm_i)$
    $cdm \leftarrow cdm \oplus cdm_i$
end for
```

Fig. 2. High-level algorithm for incremental CDM synthesis

3.2. BPM as Basis for CDM Derivation

BPMs contain some typical concepts and represent some typical facts that are inherent to business processes but may be differently represented by different modeling notations. Regardless of the applied notation, those facts and concepts possess a certain *semantic capacity* that allows the automatic CDM synthesis. Here we provide a short overview of the identified semantic capacity of BPMs for the automatic CDM synthesis (Fig. 4).[10]

[10] For the complete formal specification of the transformation rules for the direct BPM-driven CDM synthesis we refer the readers to [16,20]. Without loss of generality, source BPM concepts are represented by BPMN.

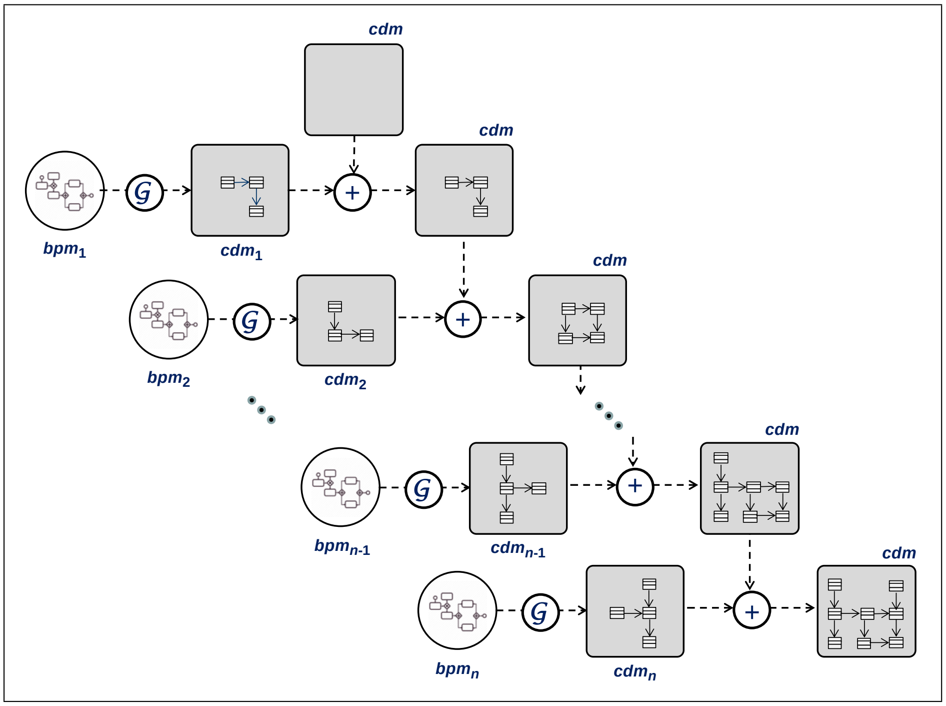


Fig. 3. Incremental CDM synthesis (binary ladder integration strategy)

The *entity types* in the target CDM can be derived from the following BPM concepts: *participants*, *roles*, *message flows*, *objects*, and *activations of existing objects*. The mapping of *participants* and their *roles* into the corresponding classes is specified by the T_1 rule (Fig. 4), while the mapping of different types of *objects* and *message flows* is specified by the T_2 rule. The T_3 rule specifies the mapping of the *activated existing objects*, i.e. objects that are created in other processes but *activated*[11] in the given process.

The *relationships* between classes in the target CDM can be automatically derived from several typical BPM patterns. The T_4 rule specifies the generation of the *participant-participant* associations between the class representing a *participant* and the classes representing its *roles*. The automatic generation of the *participant-object* associations is specified by the following rules: T_5 – *creation and subsequent usage of the generated objects*, T_6 – *exchange of messages*, and T_7 – *activation and subsequent usage of the activated objects*. The automatic generation of the *object-object* associations is specified by two transformation rules: T_8 – *association between the existing object and the corresponding activation class*, and T_9 – *tasks having input and output objects of different types*.

[11] An *activation* represents the fact that an existing object constitutes input in a task that changes its state. After activation, the *activated* object may be further used in some subsequent tasks, in the same way as generated objects are used.

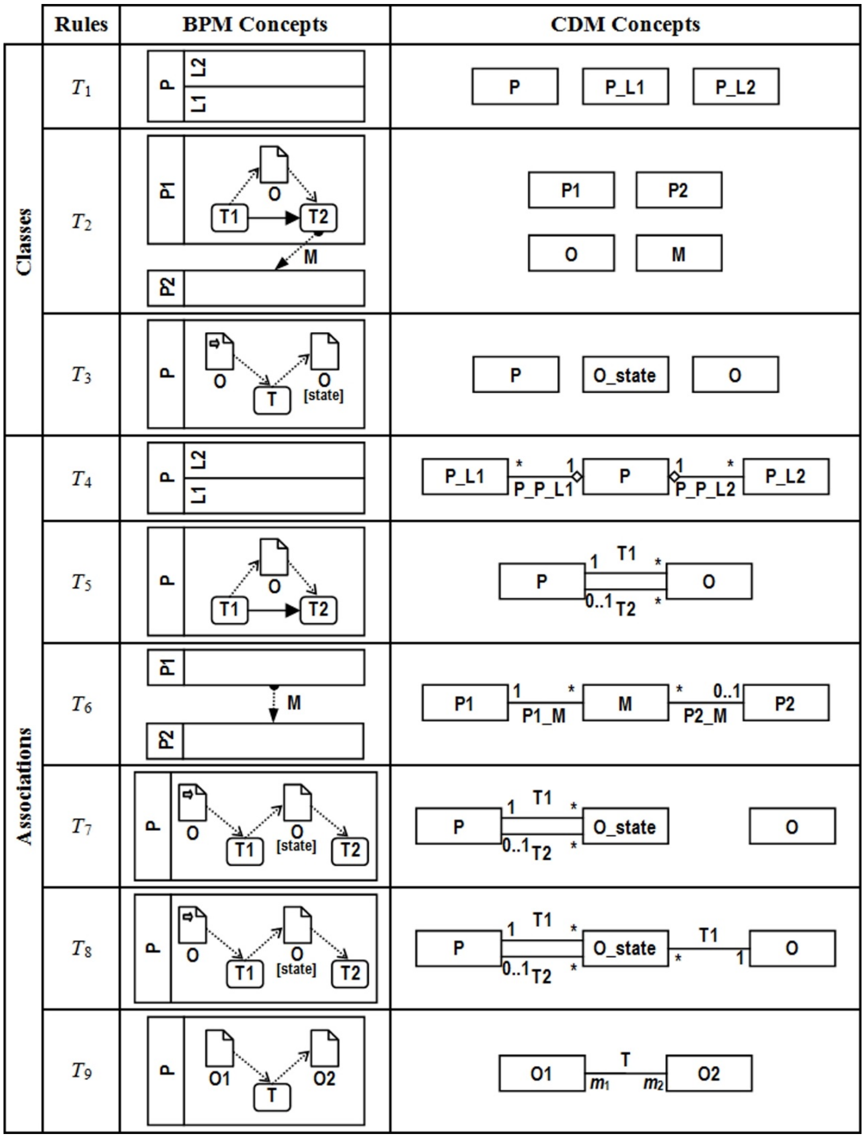


Fig. 4. Mapping of BPM concepts into CDM concepts [14]

3.3. Two-phase BPM-driven Synthesis of Partial CDMs

Each model transformation includes two types of actions: (1) *extraction* of specific elements from the source model(s), and (2) *generation* of the corresponding elements in the target model. In the existing MDSDM approaches, these two types of actions are strongly coupled and implemented by a single transformation program that takes the source BPM (represented by some concrete notation such as BPMN) and generates the target model (represented by another concrete notation, typically class diagram).

In order to overcome disadvantages of the *direct synthesis*, AMADEOS implements the *two-phase CDM synthesis* [21,19], which means that the extracting and generating actions are decoupled and separated into two consecutive activities (phases). In the first phase, appropriate *extractors* extract specific concepts from the source BPM and represent them by BMRL[12]. In the second phase, the *generator* generates the target CDM (UML class diagram) based on the BMRL-represented extracted concepts. Figure 5 illustrates the technical perspective of the approach, while Fig. 6 illustrates the two-phase CDM synthesis based on a simple BPMN model.[13] As illustrated in Fig. 5, the two-phase approach enables simple extensibility and support for other process-oriented notations (which are not necessarily metamodel-based) by implementing additional extractors.

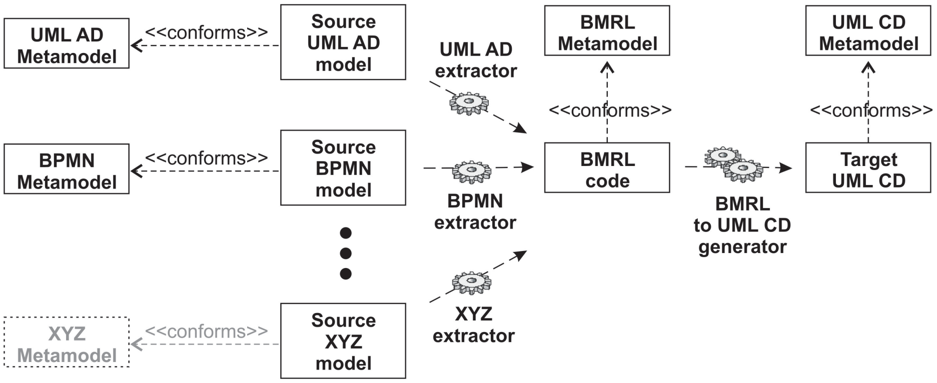
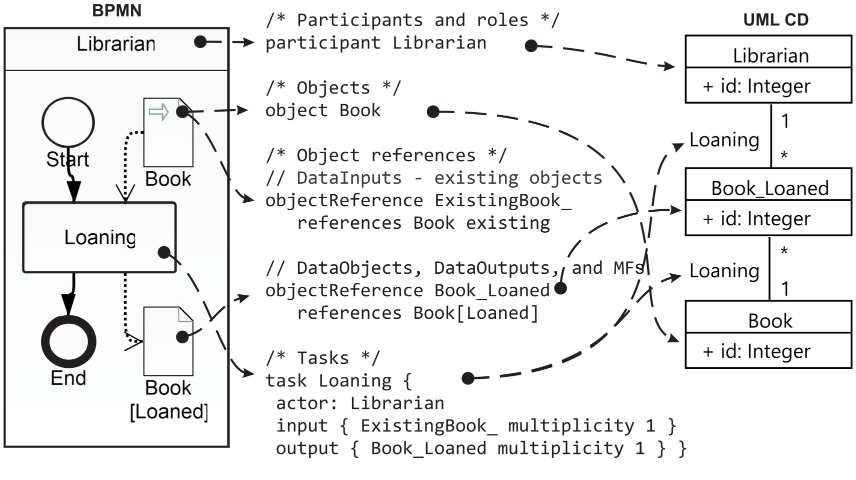


Fig. 5. Technical perspective of two-phase BPM-driven CDM synthesis [19]



```
/* Participants and roles */
participant Librarian

/* Objects */
object Book

/* Object references */
// DataInputs - existing objects
objectReference ExistingBook_
   references Book existing

// DataObjects, DataOutputs, and MFs
objectReference Book_Loaned
   references Book[Loaned]

/* Tasks */
task Loaning {
 actor: Librarian
 input { ExistingBook_ multiplicity 1 }
 output { Book_Loaned multiplicity 1 } }
```

Fig. 6. From BPM (BPMN) through BMRL to CDM (UML class diagram)

[12] BMRL (*Business Model Representation Language*) is a simple domain-specific language aimed at representing the BPM concepts having the semantic capacity for automatic CDM synthesis. For the complete specification of BMRL, we refer the readers to [21,19].

[13] For the complete formal specification of the rules for the both phases we refer the readers to [21,19].

3.4. Partial CDM Composition

The whole CDM synthesis process is performed iteratively. In each iteration, the target CDM is incrementally built by composing two partial CDMs – a CDM derived from one of the source BPMs, and a CDM obtained for already processed source BPMs. In other words, in each iteration two UML class diagrams are to be composed into a single model. The model composition is performed by applying two sets of rules (Fig. 7), whereby each group deals with specific elements.

The first group (R1) of the rules, which considers the classes and their properties, consists of the following rules:

Rule R1.1: Each class is identified with its name[14]. If both source CDMs contain the same-named classes, we conclude that they both represent the same class, and the composition result (resulting CDM) is to contain only one corresponding class of the same name. If a class contained in one of the source CDMs does not yet exist in the resulting CDM, the corresponding same-named class is to be created in the resulting CDM. If the same-named class already exists in the resulting CDM, a new class will not be created in the resulting CDM.

Rule R1.2: Each class property is identified with its name. When a new class is added to the resulting CDM, all properties are copied from the source class to the target class in the resulting CDM. If the same-named class already exists in the resulting CDM, only missing properties are to be added to the target class in the resulting CDM.

The second group (R2) of the composing rules, which deals with the classes relationships, consists of the following rules:

Rule R2.1: The association, aggregation, and composition relationships are identified by their names and the corresponding classes among which the relationships exist. If the same-named relationship exists between different classes, they are to be treated as different relationships during the composition.

Rule R2.2: The association, aggregation, and composition relationships represent the structural relationships between objects, but have different weights in the semantics that are to be considered during the model composition, i.e. $association \prec aggregation \prec composition$ (association is the weakest relationship type). If two the same-named, but differently strong, relationships exist between the same-named classes in both source CDMs, then the resulting CDM is to contain the weaker same-named relationship between the corresponding classes.

Rule R2.3: The relationship ends have the multiplicities represented by the lower (*lbv*) and upper (*ubv*) bound values ($[lbv..]ubv$, $lbv = m$, $ubv = n \vee *$, $m \in \mathbb{N}_0$, $n \in \mathbb{N}$). During the composition, lower and upper bound values of multiplicities are compared for the conflicting relationships (the same-named relationships between the same-named classes), and the most flexible bound values ($min(lbv)..max(ubv)$) are to be set for the corresponding relationship ends in the resulting CDM.

Rule R2.4: The generalization relationships do not have names. If the resulting CDM does not already contain the given generalization relationship that exists in the source CDM, the corresponding generalization relationship is to be created between the corresponding classes in the target model.

[14] Strictly speaking, each serialized model element is uniquely identified by the corresponding *id* attribute, but in the context of the model composition, the characteristic model elements are identified by their names.

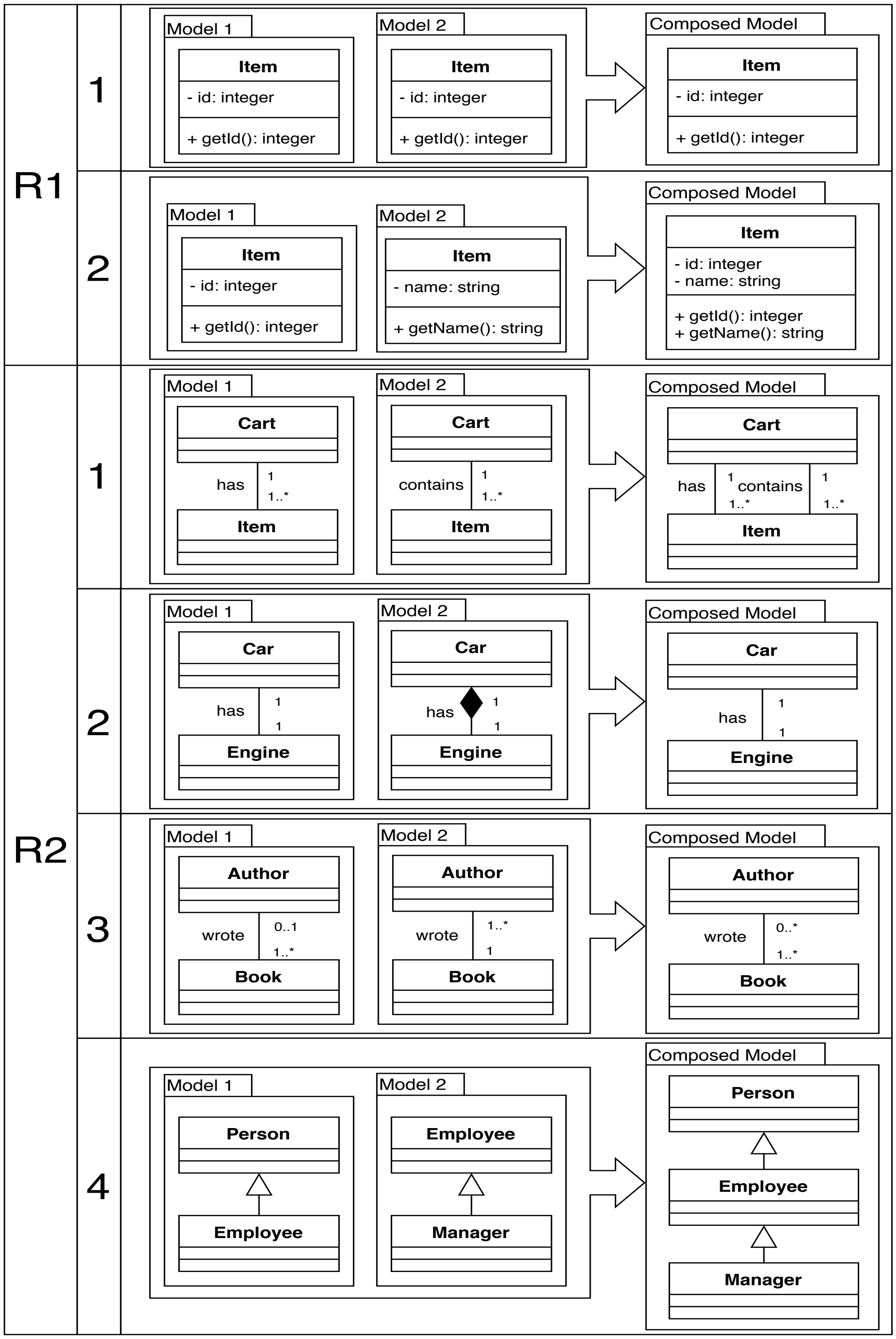


Fig. 7. Rules for composing partial CDMs

3.5. Dealing with Source Model Inconsistencies

The aforementioned model composition rules are applicable to an ideal set of the source models. However, a number of modelers are usually involved in business process modeling, and if strict modeling guidelines are not set before starting the modeling process, there is a high probability that some inconsistencies occur in the created models. These inconsistencies can be seen in the usage of different naming notations, synonyms, etc.[15]

To overcome some inconsistencies, AMADEOS provides its users a possibility to use advanced composition approach. This means that the aforementioned set of composition rules is applied, but AMADEOS additionally tries to overcome different naming notations problem and accidental typing errors. These goals are achieved by combining the following techniques: (1) case sensitivity and (2) Levenshtein distance.

Case sensitivity defines whether lowercase and uppercase letters in text are treated as distinct (case-sensitive) or equivalent (case-insensitive). Advanced composition is using case-insensitive comparison of strings during composition, which helps to overcome the problem of different naming notations.

The Levenshtein distance (LD) [50] is used[16] to overcome accidental typing errors. During advanced model composition, the measured distance between names of the model elements (i.e. the minimum number of single character insertions, deletions or substitutions required to transform one word to the other) is checked against the threshold value that is specified in the configuration file of the composer service (at the moment, threshold value is set to 1). Specified threshold value represents the number of single-character edits that are allowed. If measured distance is lower or equal to threshold value, words will be treated as equivalent.

4. Implemented Tool

The presented approach is implemented in the AMADEOS system. Figure 8 shows the architecture of the improved system. In comparison with the pre-existing version, the improved system ia able to automatically generate the target CDM based on a set of source BPMs. AMADEOS currently supports two different source notations (BPMN and UML AD), but all models in the entire source set must be represented by the same notation. Source UML AD models can be XMI-serialized, while BPMN models can be XMI- or XSD-serialized (all models in the entire set must be serialized in the same way).

The server-side is implemented as a set of web services. The *Orchestrator* service orchestrates the entire process, while each particular activity is implemented by the corresponding web service. The first phase of the CDM synthesis process is implemented by the *BPMN extractor* and *UML AD extractor* services. These two services implement the extraction of the characteristic concepts from the source BPM, and generation of the corresponding BMRL code. The second phase of the CDM synthesis process is implemented by the *UML CD generator* service, while the model composition is performed by the *UML CD composer* service. Finally, the *UML CD layouter* service is aimed at automatic layout of the corresponding UML class diagram.

[15] For the most comprehensive taxonomy of merging conflicts we refer the readers to [60].

[16] In order to eliminate inconsistencies in the source BPMs, various techniques can be applied. AMADEOS currently applies a common technique based on LD. Some other algorithms could be used as well, like Jaaro-Winkler [74], which could be a part of future work.

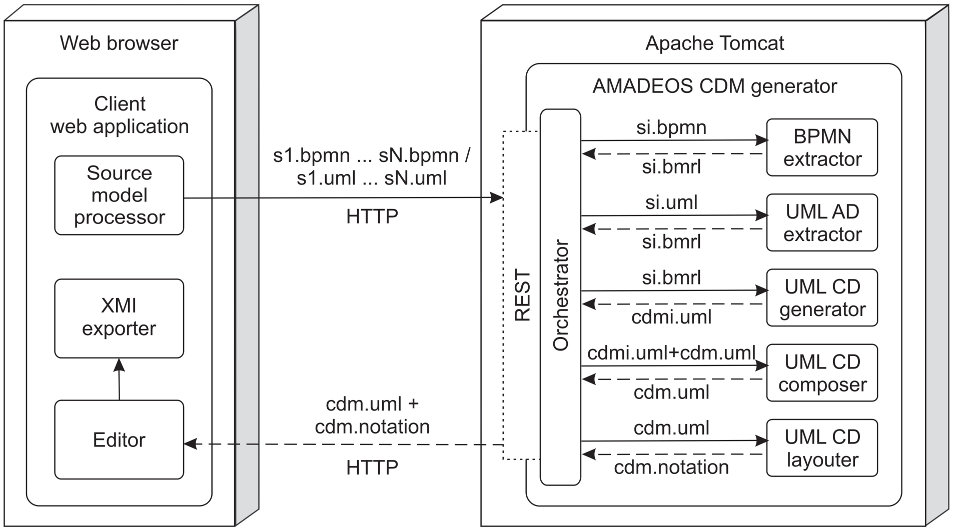


Fig. 8. Architecture of AMADEOS system

In a positive usage scenario[17] (Fig. 9), the orchestrator service receives a set of the source BPMs (*s1.bpmn...sN.bpmn/s1.uml...sN.uml*), and returns the corresponding automatically generated CDM (*cdm.uml+cdm.notation*). For each source BPM, the orchestrator orchestrates the two-phase synthesis process that results with the corresponding partial CDM, and incrementally builds the target CDM. More concretely, for each source model (*si.bpmn/si.uml*), the orchestrator firstly sends it to the corresponding extractor service, which generates and returns the corresponding BMRL code (*si.bmrl*). The orchestrator further sends the BMRL code to the generator service that generates and returns the corresponding partial CDM (*cdmi.uml*). This partial CDM and the CDM obtained for already processed source BPMs, are further sent (*cdmi.uml+cdm.uml*) to the composer service that performs model composition and returns the composed CDM (*cdm.uml*). After processing all source BPMs, the orchestrator forwards the resulting CDM (*cdm.uml*) to the layouter service, which automatically generates and returns the layout of the corresponding diagram (*cdm.notation*). Finally, the model and the diagram are merged by the orchestrator into a single JSON[18] object (*cdm.uml+cdm.notation*), and returned to the client.

The *client web application* (Fig. 10) allows users to upload a set of source BPMs to the orchestrator service. When the entire synthesis process is finished, the client application receives the JSON response and visualizes[19] the class diagram in the browser (*Editor* component). The visualized diagram is editable so users can additionally improve it. It is also possible to export the model in the XMI format (*XMI exporter*), and further use it in some other modeling tool.

[17] This assumes that a user has selected appropriate options in the user interface and uploaded an appropriate collection of BPMs.

[18] JavaScript Object Notation

[19] The implementation is based on the jsUML2 library (*http://www.jrromero.net/tools/jsUML2*).

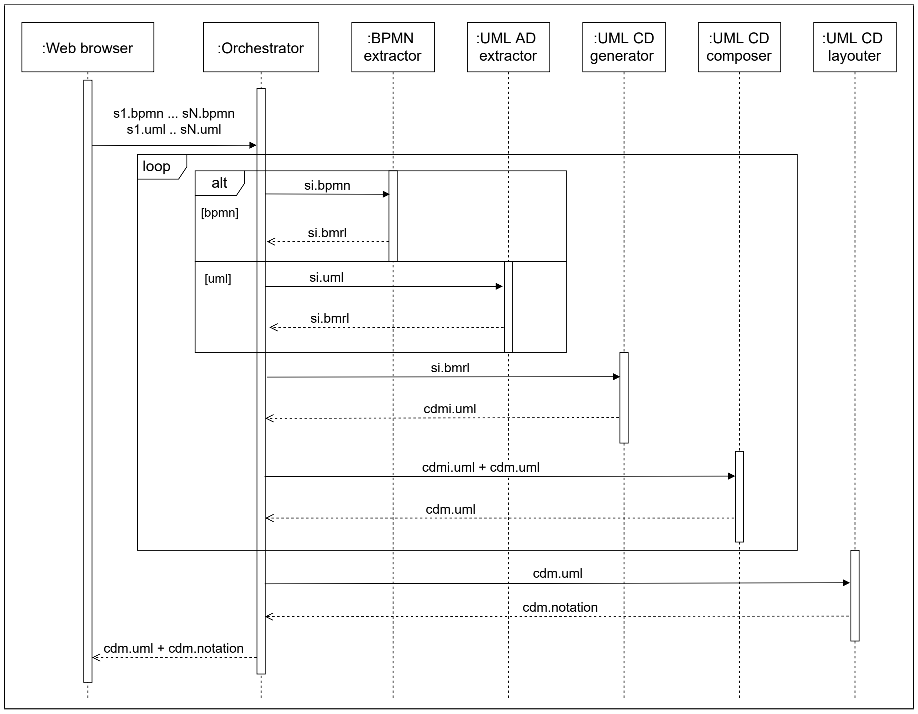
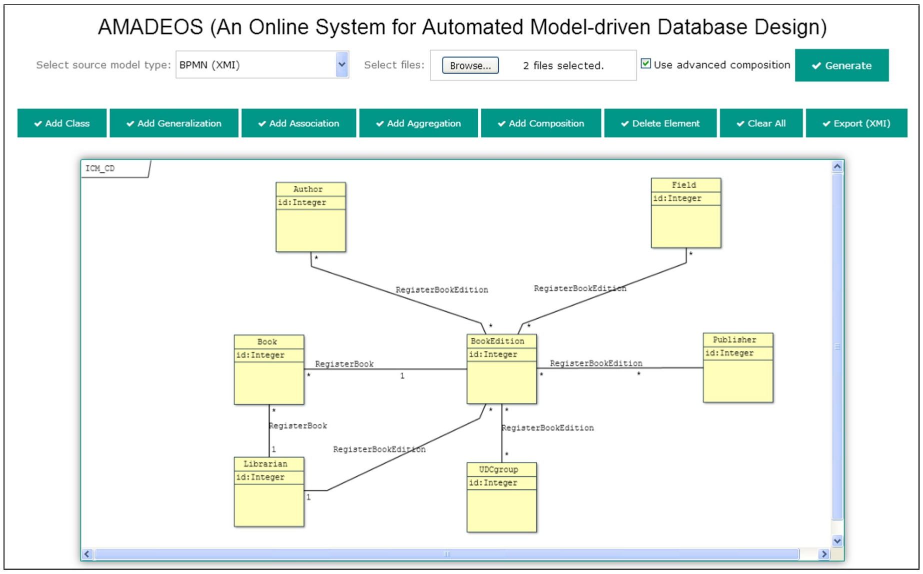


Fig. 9. Sequence diagram representing positive usage scenario of AMADEOS system



AMADEOS (An Online System for Automated Model-driven Database Design)

Select source model type: BPMN (XMI)

Select files: Browse... 2 files selected.

☑ Use advanced composition

Fig. 10. Screenshot of AMADEOS client web application

5. Illustrative Example

In this section we provide an example that illustrates the implemented approach. This example is provided here for the purpose of the approach illustration, while the next section presents the results of the experimental evaluation of the entire approach.

The sample source set of BPMs (Fig. 11), which is used in this example, represents three main processes in the Faculty Library: (1) Book Edition Registration, (2) Book Registration, and (3) Book Write Off. These three simple, mutually related models belong to the set of BPMs of the Faculty Library, which is used for evaluation (next section) of the implemented approach.

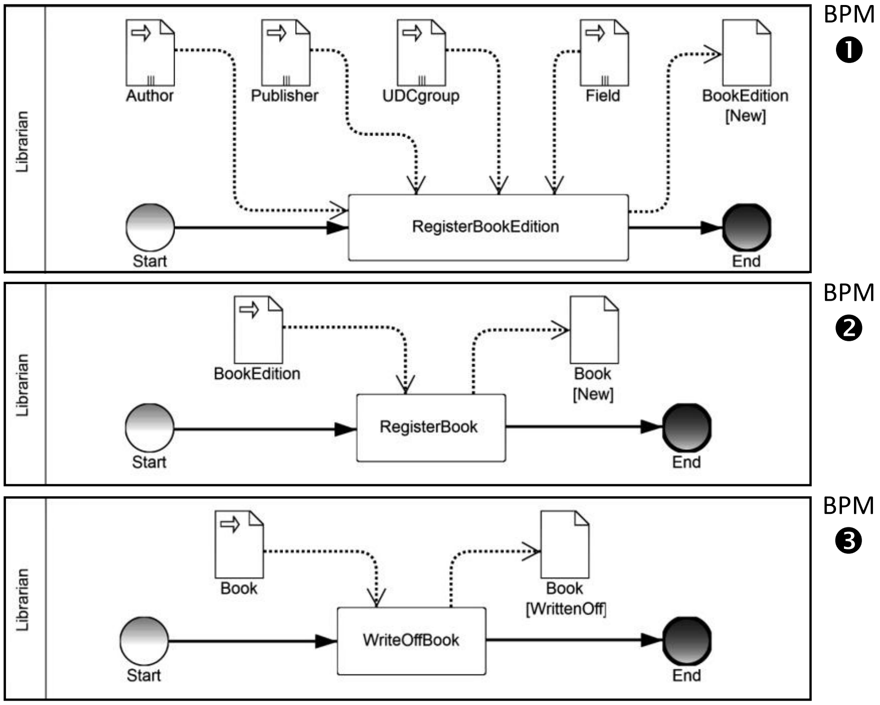


Fig. 11. Sample source set of BPMN models

Figure 12 illustrates the process of the incremental CDM composition based on the sample source set of BPMs. Firstly, CDM Generator generates the corresponding CDM based on the first BPM (Book Edition Registration). After that, CDM Generator derives the corresponding CDM from the second BPM (Book Registration), and then UML CD Composer composes these two partial CDMs into the resulting CDM. Finally, UML CD Composer combines this resulting CDM with the CDM derived from the third BPM (Book Write Off), and generates the CDM that corresponds to the sample source set.[20]

[20] Figure 12 contains class diagrams that correspond to the automatically generated CDMs in the AMADEOS system, after some manual improvements of the automatically generated layout.

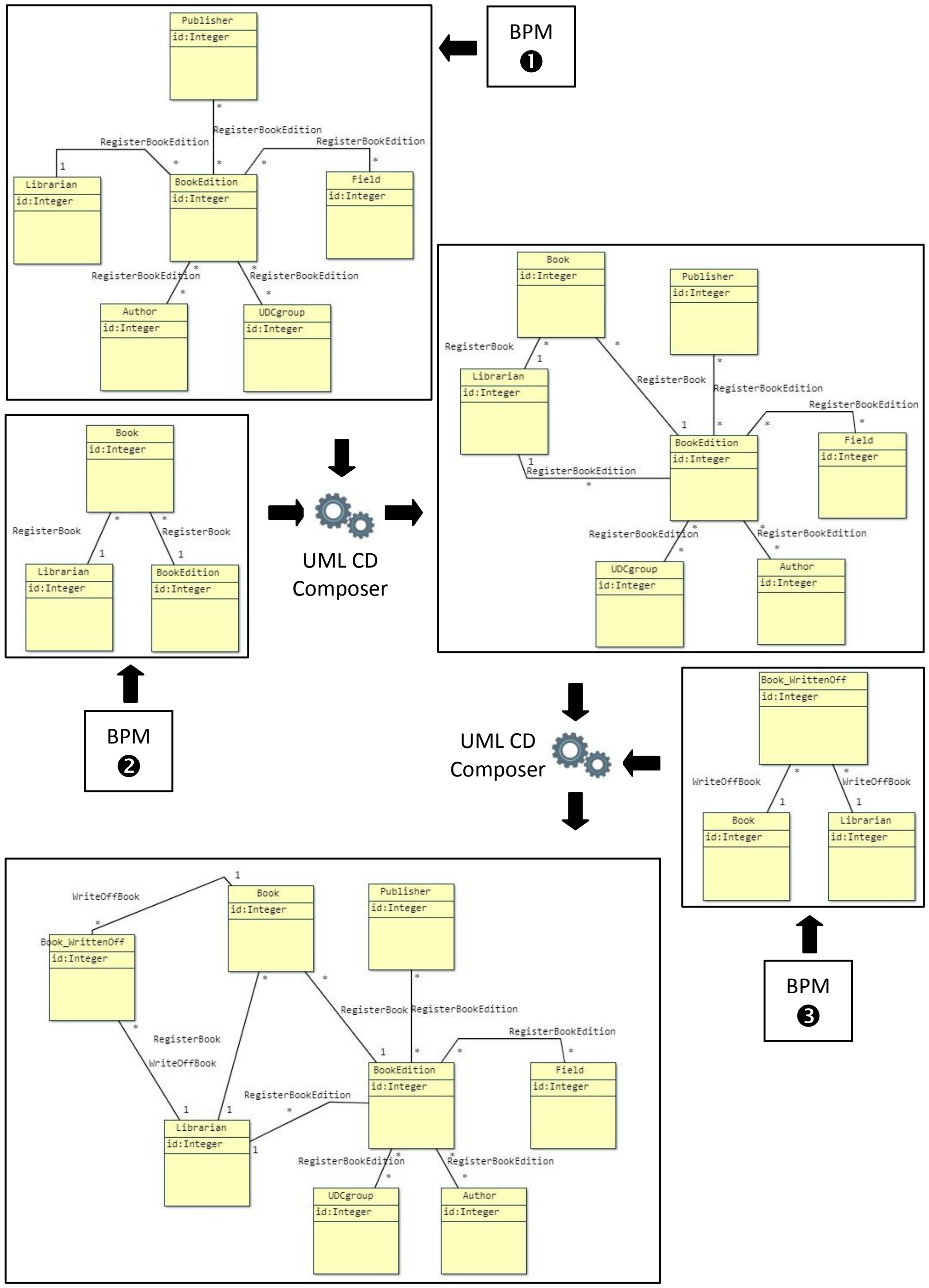


Fig. 12. Illustration of incremental CDM composition based on sample source set of BPMN models

6. Evaluation

In order to evaluate the implemented approach, we conducted a very extensive evaluation focused on the approach effectiveness, through the assessment of correctness and completeness of the CDMs automatically derived from the sets of BPMs. Firstly, we performed a case study-based evaluation. After that, we performed several experiments in order to verify the obtained case study-based results. This section presents these evaluation activities and obtained results – firstly the case study, followed by the experiments. Finally, we consider threats to validity of the experiments and derived conclusions, as well as lessons we learned.

6.1. Case Study

The implemented approach has been firstly evaluated through a case study of the Faculty Library Information System, through the assessment of the CDM, which is automatically derived from a set of BPMs of the Faculty Library, with a CDM that corresponds to the Library database.

Reference models. Figure 13 depicts a class diagram representing the CDM that corresponds to the existing (current) Library database within the Faculty Library Information System. The given class diagram is manually designed based on the corresponding relational database schema.[21] In the rest of the article, we use the *reference CDM* term for this model, since it is used later as a reference in the experiments.

In order to evaluate the approach and implemented tool, we used a set of BPMs of the Faculty Library. This collection contains 14 models[22], some of which are already shown in the example illustrating the incremental composition process in the previous section. The labels and names of the corresponding business processes are shown in Table 1. The used collection contains seven models of *main* (operational/transactional) *processes* (R-01...R-07) and seven models of *auxiliary processes* (R-08...R-14). The main processes allow the creation of new data (e.g. R-01 represents the process of registering a new book edition, R-02 represents the process of registering a new book, etc.). The auxiliary processes enable the maintenance of the existing registers and the modification of the existing data (e.g. R-08 represents the process of modifying authors' data, etc.). This collection of BPMN models was also (partly) used in the experiments and is hereinafter referred to as the *reference set of BPMs*.

Evaluation results. Based on the reference set of BPMs, CDMs were automatically generated using the AMADEOS tool, and then the generated CDMs were compared with the reference CDM of the Faculty Library Information System (Fig. 13). We were particularly interested in the contribution of the *models of the main processes* (*main BPMs* in the rest of the article) and the contribution of the *models of the auxiliary processes* (*auxiliary BPMs* in the rest of the article). Therefore, we generated CDMs based on each reference BPMN model, as well as based on the partial reference set containing only the main BPMs, and the complete reference set of BPMs.[23]

[21] Note that class attributes are left out because at the moment AMADEOS is not able to generate class attributes, except a single attribute for each class, which represents its primary key.

[22] Available at: *https://gitlab.com/m-lab-research/amadeos-exp-2020/-/tree/master/CS/Ref-BPMs*

[23] The generated models are available at: *https://gitlab.com/m-lab-research/amadeos-exp-2020/-/tree/master/CS/CDMs*

Table 1. Reference BPMN models of main (left) and auxiliary processes (right)

Main processes		Auxiliary processes	
ID	Process name	ID	Process name
R-01	Book Edition Registration	R-08	Author Update
R-02	Book Registration	R-09	Book Edition Update
R-03	Member Enrollment	R-10	Publisher Update
R-04	Member Seceding	R-11	Book Update
R-05	Book Borrowing	R-12	Field Update
R-06	Book Returning	R-13	UDC Group Update
R-07	Book Write Off	R-14	Member Record Update

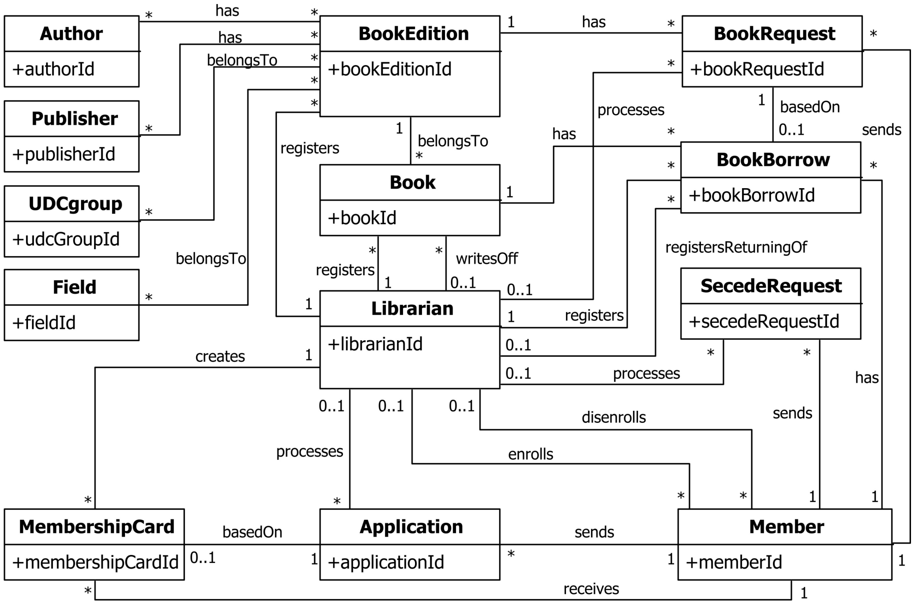


Fig. 13. Reference CDM of Faculty Library Information System

We use the following metrics for the quantitative evaluation of the generated CDM in comparison with the reference CDM:

- N_g – total number of generated concepts in the generated CDM,
- N_c – number of correctly generated concepts in the generated CDM, i.e. number of concepts that are identical to the concepts contained in the reference CDM,
- N_w – number of incorrectly generated concepts in the generated CDM, i.e. number of concepts that are not present in the reference CDM, and
- N_m – number of missing concepts in the generated CDM in comparison with the reference CDM.

We use *recall*, *precision*, and *F-score* as measures for the evaluation of the automatically generated CDM.

Recall (R) constitutes the measure of *completeness* of the generated CDM in comparison to the reference CDM. It may be defined as follows:

$$R = \frac{N_c}{N_c + N_m}. \tag{1}$$

Precision (P) constitutes the measure of *correctness* of the generated CDM. It may be defined as follows:

$$P = \frac{N_c}{N_c + N_w}. \tag{2}$$

F-score (F) constitutes the *effectiveness* measure. It represents the harmonic mean of precision (P) and recall (R), and it may be defined as follows:

$$F = \frac{2PR}{P + R}. \tag{3}$$

The results of the comparison of automatically generated CDMs with the reference CDM are given in Table 2. The first 14 rows (rows labeled from ”01” to ”14”, where the label corresponds to the source model ID) contain results for the CDMs derived from the corresponding single BPMN model contained in the reference set of BPMs. The row labeled with ”01-07” contains results for the CDM derived from the partial set of the main BPMs, while the row labeled with ”01-14” contains results for the CDM derived from the complete reference set of BPMs. Figure 14 shows a comparison of R, P, and F for the generated classes (top) and associations (bottom) based on the reference BPMN models in comparison with the reference CDM.

Results discussion. The comparison results of the generated CDMs with the reference CDM show that none of the individual source BPMN models from the reference set has the semantic capacity to enable the generation of neither all classes nor all associations that exist in the reference CDM. As expected, a set of BPMs enables the generation of a more complete CDM with respect to the individual BPMs – no single BPM has enabled the generation of more than 46% of classes neither more than 20% of associations contained in the target CDM, while the set of BPMs enabled the generation of 100% of classes and 72% of associations contained in the target CDM.

Regarding the generation of classes, the results show that AMADEOS generated all classes that exist in the reference CDM, when the source reference set of BPMs was used, and also show that only main BPMs were sufficient to generate all classes (R=1 for the CDM derived from the set containing models R-01...R-07). Based on the auxiliary BPMs, additional classes were generated in comparison to the reference CDM, which constitute a surplus in this particular case.[24] These excessive classes, which were generated based on the auxiliary BPMs, do not increase the recall but reduce the precision because they represent a surplus in this particular case. The achieved precision in generating the classes is P=0.65 (based on the set of the main BPMs), and P=0.48 (based on the complete reference set). Since recall in both cases is R=1, the *F-score* for the class generation process is F=0.79 (based on the set of the main BPMs), and F=0.65 (based on the complete reference set).

[24] The analysis of excessive classes shows that these classes represent activation classes, which are not important in this case, because the observed system does not keep a history of changes in the state of individual objects, but only records their current states/values.

Table 2. Assessment of automatically generated CDMs (based on reference BPMN models) in comparison with reference CDM

Source model(s)	Assessment of classes							Assessment of associations						
	N_g	N_c	N_m	N_w	R	P	F	N_g	N_c	N_m	N_w	R	P	F
01	6	6	7	0	0.46	1.00	0.63	5	5	20	0	0.20	1.00	0.33
02	3	3	10	0	0.23	1.00	0.38	2	2	23	0	0.08	1.00	0.15
03	6	4	9	2	0.31	0.67	0.42	13	4	21	9	0.16	0.31	0.21
04	7	3	10	4	0.23	0.43	0.30	16	2	23	14	0.08	0.13	0.10
05	7	5	8	2	0.38	0.71	0.50	15	5	20	10	0.20	0.33	0.25
06	3	1	12	2	0.08	0.33	0.13	2	0	25	2	0.00	0.00	N/A
07	3	2	11	1	0.15	0.67	0.25	2	0	25	2	0.00	0.00	N/A
08	3	2	11	1	0.15	0.67	0.25	2	0	25	2	0.00	0.00	N/A
09	3	2	11	1	0.15	0.67	0.25	2	0	25	2	0.00	0.00	N/A
10	3	2	11	1	0.15	0.67	0.25	2	0	25	2	0.00	0.00	N/A
11	3	2	11	1	0.15	0.67	0.25	2	0	25	2	0.00	0.00	N/A
12	3	2	11	1	0.15	0.67	0.25	2	0	25	2	0.00	0.00	N/A
13	3	2	11	1	0.15	0.67	0.25	2	0	25	2	0.00	0.00	N/A
14	3	2	11	1	0.15	0.67	0.25	2	0	25	2	0.00	0.00	N/A
01-07	**20**	**13**	**0**	**7**	**1.00**	**0.65**	**0.79**	**55**	**18**	**7**	**37**	**0.72**	**0.33**	**0.45**
01-14	27	13	0	14	1.00	0.48	0.65	69	18	7	51	0.72	0.26	0.38

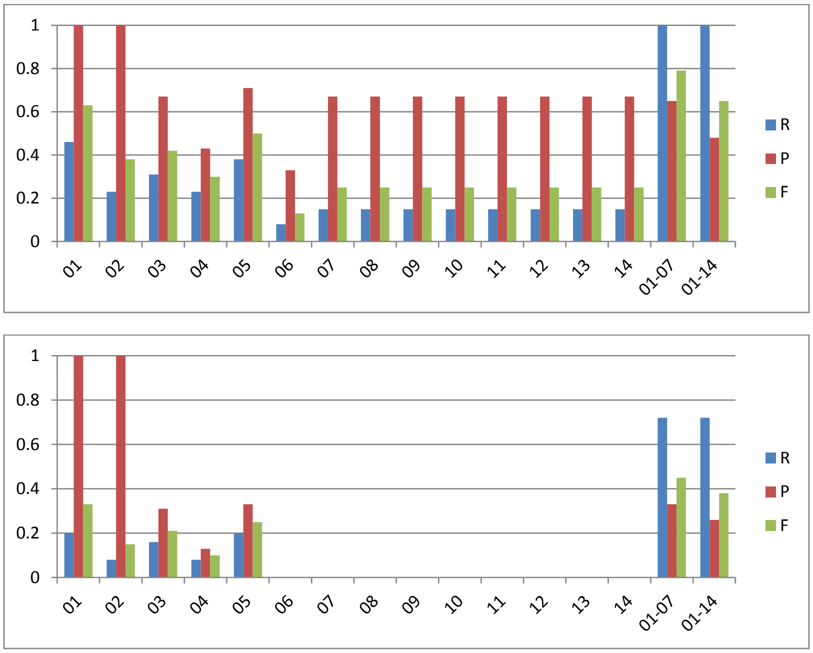


Fig. 14. Comparison of *R*, *P*, and *F* for generated classes (top) and associations (bottom) based on reference BPMN models in comparison with reference CDM

Regarding the generation of associations, the results show that AMADEOS generated 72% of associations that exist in the reference CDM, when the source reference set of BPMs was used, and also show that this recall can be achieved by using only main BPMs (R-01...R-07) as a basis for the generation. In this case, the auxiliary BPMs did not contribute to the recall increase, as well.

Additional analysis of the obtained results and reference set of BPMs showed that it would be possible to achieve a slightly higher recall than R=0.72, if some BPMs were further improved, but these additional improvements would not allow the generation of all associations between classes compared to the reference CDM. In other words, additional improvements to the reference BPMs cannot result in achieving R=1.00 in the associations generation process, because the tool generates associations between inappropriate classes in some situations – this refers to existing objects that are activated in one process and further used in another process. This problem does not exist when generating a CDM based on a single BPM, but it does occur when integrating partial CDMs that are generated based on a set of BPMs. This problem will be the subject of our future research, in order to increase the completeness of the automatically generated CDM.

The results show that a significant number of generated associations is excessive compared to the reference CDM, so the precision is relatively low (P=0.33 for the CDM derived from the set of the main BPMs, and P=0.26 for the CDM derived from the complete reference set). Given the relatively low precision, the effectiveness of the association generation process is F=0.45 (based on the main BPMs) and F=0.38 (based on the complete reference set). The analysis of excessively generated associations shows that the vast majority of these associations are not incorrectly generated (they have correct end multiplicities), but they do not exist in the reference CDM and therefore represent a surplus in this particular case. In other words, the given BPMs contain some process patterns having the semantic capacity for automatic generation of associations (as shown in the previous research), but these associations represent a surplus in this particular case. The problem of excessive associations (although correctly generated) has been identified in the previous research [20]. This is not a major problem when only one BPM is taken as a starting point, but with larger sets of BPMs the total number of excessively generated associations is significant, which may constitute a problem.

Since the case study showed that the main BPMs are sufficient to generate the target CDM, i.e. the auxiliary BPMs do not contribute to the increase of recall and precision, only main BPMs were further used in the experiments (described in the following subsections).

6.2. Experiment #1 (E-1)

Experiment design. According to Conway's law [28], the independent work of different designers will result in different models. Therefore, this experiment aimed to assess the effectiveness of the implemented approach in generating the CDM based on different real sets of BPMs, i.e. how real models affect the effectiveness of the generation process.

A total of 98 undergraduate students participated in E-1.[25] The experiment was conducted as a part of a mandatory course that includes business process modelling. All

[25] All participants in E-1 were students of the third year at the Software Engineering Department, at the Faculty of Electrical Engineering, University of Banja Luka.

participants firstly underwent the usual training (10 hours) for the BPMN-based business process modelling using the appropriate modelling tool (Eclipse/BPMN Modeler). Upon completion of the training, the task was to create a set of BPMN models based on the given textual specification of the main business processes in the Faculty Library. Given the scope of work and available time, all participants performed the task in pairs, which resulted in a total of 49 sets of the main BPMs of the Faculty Library. Given the previous education, as well as level of knowledge and skills in the BPMN-based business process modelling, all participants can be considered novice modelers.

After the analysis and evaluation of the created sets of the main BPMs[26], we have singled out 27 sets that were graded with a minimum of 70/100. These sets[27] were used as a starting point for the automatic CDM generation in the AMADEOS system. Based on each set, two CDMs were generated: (1) with applied ”Advanced Composition” option[28] (case insensitivity and *LD*=1), and (2) without the ”Advanced Composition” option[29] (case sensitivity and *LD*=0). The generated CDMs were then compared with the reference CDM. When evaluating the generated CDMs against the reference CDM, we used the same metrics and measures as in the case study.

Results. The comparison results of the automatically generated CDMs (”Advanced Composition” applied), with the reference CDM, are shown in Table 3. Each table row contains results for the CDM generated based on the corresponding set of the BPMN models (the first column contains the ID of the source set of the BPMN models, i.e. ID of the corresponding automatically generated CDM), while the other columns contain the corresponding metrics and measures for automatically generated classes and automatically generated associations, respectively. Figure 15 shows a comparison of R, P, and F for the generated classes (top) and associations (bottom) based on the students’ sets of the BPMN models in comparison with the reference CDM.

Discussion. The experimental results largely confirm the results obtained in the case study.

Regarding the generation of classes, the obtained results confirm that a set of the main BPMs enables the generation of a very complete CDM because a very high average recall was achieved ($\overline{R}$=0.94) when using 27 real sets of the BPMN models ($\overline{R}$=1.00 in the case study). The achieved average precision for the class generation is $\overline{P}$=0.59 ($\overline{P}$=0.65 in the case study), and the average effectiveness of the class generation process is $\overline{F}$=0.72 ($\overline{F}$=0.79 in the case study).

Regarding the generation of associations, the average completeness is $\overline{R}$=0.57 ($\overline{R}$=0.72 in the case study) with the average precision $\overline{P}$=0.29 ($\overline{P}$=0.33 in the case study). This means that the average effectiveness of generating associations is $\overline{F}$=0.38 ($\overline{F}$=0.45 in the case study). The high matching of the achieved precision in the experiment ($\overline{P}$=0.29) and the case study ($\overline{P}$=0.33) confirms that the generator generates a significant number of excessive associations (relative to the reference CDM).

[26] The analysis and evaluation were conducted by the teachers of the course, who are also some of the coauthors of this article.

[27] Available at: *https://gitlab.com/m-lab-research/amadeos-exp-2020/-/tree/master/E-1/BPMs*

[28] Available at: *https://gitlab.com/m-lab-research/amadeos-exp-2020/-/tree/master/E-1/CDMs-AC*

[29] Available at: *https://gitlab.com/m-lab-research/amadeos-exp-2020/-/tree/master/E-1/CDMs-NoAC*

Table 3. Assessment of automatically generated CDMs (”Advanced Composition” applied), based on students’ sets of BPMN models, in comparison with reference CDM

Source set	Assessment of classes							Assessment of associations						
	N_g	N_c	N_m	N_w	R	P	F	N_g	N_c	N_m	N_w	R	P	F
01	16	12	1	4	0.92	0.75	0.83	49	14	11	35	0.56	0.29	0.38
02	20	12	1	8	0.92	0.60	0.73	53	11	14	42	0.44	0.21	0.28
03	18	13	0	5	1.00	0.72	0.84	43	17	8	26	0.68	0.40	0.50
04	16	12	1	4	0.92	0.75	0.83	47	16	9	31	0.64	0.34	0.44
05	19	13	0	6	1.00	0.68	0.81	41	17	8	24	0.68	0.41	0.52
06	20	12	1	8	0.92	0.60	0.73	50	14	11	37	0.56	0.27	0.37
07	19	12	1	7	0.92	0.63	0.75	40	14	11	26	0.56	0.35	0.43
08	20	11	2	9	0.85	0.55	0.67	57	15	10	42	0.60	0.26	0.37
09	24	13	0	11	1.00	0.54	0.70	67	18	7	49	0.72	0.27	0.39
10	27	13	0	14	1.00	0.48	0.65	62	12	13	50	0.48	0.19	0.28
11	13	13	0	10	1.00	0.57	0.72	39	16	9	23	0.64	0.41	0.50
12	18	13	0	5	1.00	0.72	0.84	46	16	9	30	0.64	0.35	0.45
13	31	13	0	18	1.00	0.42	0.59	69	14	11	55	0.56	0.20	0.30
14	18	12	1	6	0.92	0.67	0.77	54	17	8	37	0.68	0.31	0.43
15	20	12	1	8	0.92	0.60	0.73	36	14	11	22	0.56	0.39	0.46
16	21	13	0	8	1.00	0.62	0.76	52	16	9	36	0.64	0.31	0.41
17	27	13	0	14	1.00	0.48	0.65	60	15	10	45	0.60	0.25	0.35
18	24	13	0	11	1.00	0.54	0.70	69	13	12	56	0.52	0.19	0.28
19	23	12	1	11	0.92	0.52	0.67	55	15	10	40	0.60	0.27	0.38
20	28	12	1	16	0.92	0.43	0.59	76	12	13	64	0.48	0.16	0.24
21	21	11	2	10	0.85	0.52	0.65	50	13	12	37	0.52	0.26	0.35
22	19	12	1	7	0.92	0.63	0.75	34	11	14	23	0.44	0.32	0.37
23	24	12	1	12	0.92	0.50	0.65	50	13	12	37	0.52	0.26	0.35
24	23	11	2	12	0.85	0.48	0.61	58	12	13	46	0.48	0.21	0.29
25	20	12	1	8	0.92	0.60	0.73	42	14	11	28	0.56	0.33	0.42
26	21	12	1	9	0.92	0.57	0.71	50	13	12	37	0.52	0.26	0.35
27	19	12	1	7	0.92	0.63	0.75	37	14	11	23	0.56	0.38	0.45
Mean	**21.4**	**12.3**	**0.7**	**9.1**	**0.94**	**0.59**	**0.72**	**51.4**	**14.3**	**10.7**	**37.1**	**0.57**	**0.29**	**0.38**

Based on the results, we can conclude that the sets of the main BPMs, although they were created by novice modelers, represent a very good starting point for generating the target CDM, because the average completeness of the generated CDM is 94% for classes and 57% for associations. This further implies that we can expect that ”each” real set of the main BPMs, created by experienced modelers, will enable the automatic generation of a more complete CDM in comparison to the recall achieved in E-1.

Impact of accidental errors on the process effectiveness. Regarding the impact of accidental errors in the source BPMs on the effectiveness of the CDM generation process, the analysis shows that the applied option ”Advanced Composition” allows to reduce the total number of generated concepts – for all 27 sets the same or fewer concepts is

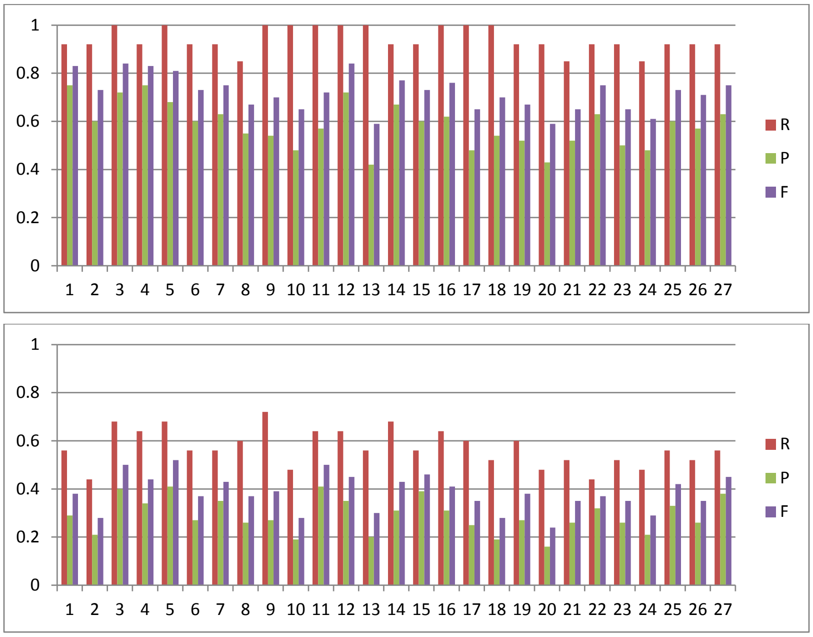


Fig. 15. Comparison of *R*, *P*, and *F* for generated classes (top) and associations (bottom) based on students' sets of BPMN models in comparison with reference CDM

generated than when the option ”Advanced Composition” is not applied. The analysis of the generated CDMs shows that in 44% (12/27) cases a reduction in the number of generated concepts was achieved, while in 56% (15/27) cases identical models were obtained. The analysis shows that the differences in the generated models are mainly related to the generated classes – in all 12/27 cases, a smaller number of classes was generated, while only in 2/27 (7%) cases a smaller number of associations was generated.

The analysis shows that the reduced number of the generated concepts did not reduce the completeness of the generated CDM compared to the reference CDM in any case (average recall is the same regardless of whether the ”Advanced Composition” option is applied). A smaller number of generated concepts reduces redundancy and increases precision (for classes: $\overline{P}_{AC}$=0.5857, $\overline{P}_{NoAC}$=0.5704; for associations: $\overline{P}_{AC}$=0.2910, $\overline{P}_{NoAC}$=0.2906). Increasing the precision with the same recall resulted in higher effectiveness (for classes: $\overline{F}_{AC}$=0.7183, $\overline{F}_{NoAC}$=0.7068; for associations: $\overline{F}_{AC}$=0.3822, $\overline{F}_{NoAC}$=0.3818). For the purpose of illustration, Fig. 16 shows differences between *P* values for the generated classes, based on the students' sets of the BPMN models in comparison with the reference CDM, in case that ”Advanced Composition” is applied (AC) or not applied (NoAC).

To conclude, applied ”Advanced Composition” reduces impact of the accidental errors and inconsistencies in the source BPMs on the effectiveness of the CDM generation process.

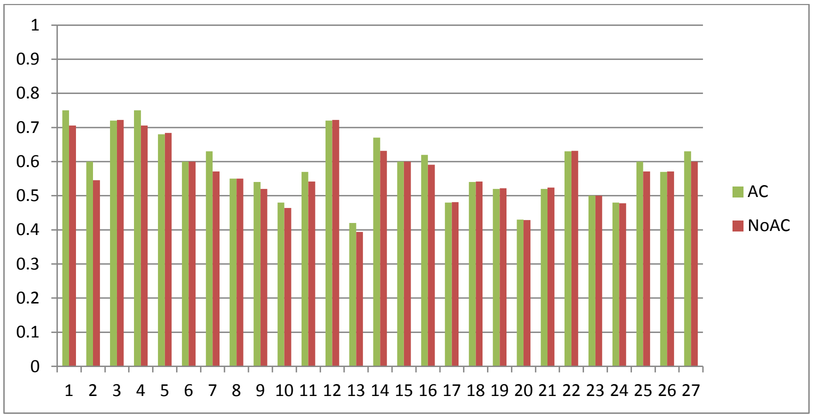


Fig. 16. Comparison of *P* values for generated classes: (AC) – ”Advanced Composition” applied, (NoAC) – ”Advanced Composition” not applied

6.3. Experiment #2 (E-2)

Experiment design. Considering Conway’s law [28], E-2 was conducted in order to assess how much the CDM, which is automatically generated based on the reference set of the main BPMs, matches real CDMs that are manually designed for the same business system, based on the same textual specifications of business processes.

A total of 18 graduate students participated in E-2.[30] The experiment was conducted as a part of an elective course that includes advanced database design techniques. All participants had the necessary knowledge and skills for manual database design, which they acquired as undergraduate students, so they did not have additional training. The task was to manually design the CDM based on the given textual specification of the main business processes in the Faculty Library (the same specification was given to the participants in E-1 for creating the BPMN models). The participants solved the task in pairs. In the end we had a total of nine manually designed CDMs for the Faculty Library.[31] All manually designed CDMs were analyzed by the teachers and evaluated as adequate.[32] Each manually designed CDM was then compared with the CDM automatically generated based on the reference set of the main BPMs, and the same metrics and measures were used in the evaluation as in the case study and in E-1.

Results. The results of comparing the CDM, which was automatically generated based on the reference set of the main BPMs, with the manually designed CDMs are shown in Table 4. Each table row contains results for the corresponding manually designed CDM – the first column contains the ID of the manually designed CDM, while the other columns contain the corresponding metrics and measures for automatically

[30] All participants in E-2 were master students at the Software Engineering Department, at the Faculty of Electrical Engineering, University of Banja Luka.

[31] Available at: *https://gitlab.com/m-lab-research/amadeos-exp-2020/-/tree/master/E-2/CDMs*

[32] The analysis and assessment were conducted by the teachers of the course, who are also some of the coauthors of this article.

generated classes and automatically generated associations, respectively. The N-labeled column contains a number of the corresponding concepts in the real CDM. Figure 17 shows a comparison of R, P, and F for the classes and associations in the automatically generated CDMs (based on the reference set of the main BPMs) in comparison with the manually designed CDMs.

Table 4. Assessment of automatically generated CDM (based on reference set of main BPMs) in comparison with manually designed CDMs

CDM	Assessment of classes								Assessment of associations							
ID	N	N_g	N_c	N_m	N_w	R	P	F	N	N_g	N_c	N_m	N_w	R	P	F
01	10	20	10	0	10	1.00	0.50	0.67	12	55	9	3	46	0.75	0.16	0.27
02	13	20	12	1	8	0.92	0.60	0.73	21	55	10	11	45	0.48	0.18	0.26
03	10	20	10	0	10	1.00	0.50	0.67	21	55	16	5	39	0.76	0.29	0.42
04	15	20	13	2	7	0.87	0.65	0.74	20	55	12	8	44	0.60	0.22	0.32
05	16	20	15	1	5	0.94	0.75	0.83	20	55	13	7	42	0.65	0.24	0.35
06	16	20	15	1	5	0.94	0.75	0.83	24	55	12	12	43	0.50	0.22	0.30
07	9	20	8	1	12	0.89	0.40	0.55	8	55	5	3	50	0.63	0.09	0.16
08	10	20	10	0	10	1.00	0.50	0.67	9	55	7	2	48	0.78	0.13	0.22
09	18	20	18	0	2	1.00	0.90	0.95	35	55	23	12	32	0.66	0.42	0.51
Mean	**13**	**20**	**12.3**	**0.7**	**7.7**	**0.95**	**0.62**	**0.74**	**19**	**55**	**11.9**	**7.1**	**43.1**	**0.64**	**0.22**	**0.31**

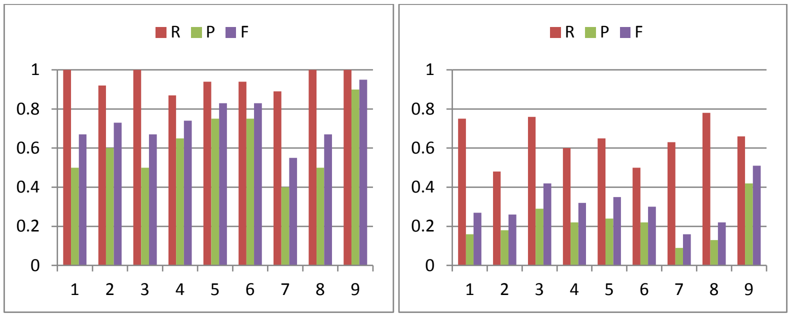


Fig. 17. Comparison of R, P, F for generated classes (left) and associations (right) in automatically generated CDMs (based on reference set of main BPMs) in comparison with manually designed CDMs

Discussion. Both the experimental results achieved in E-2, as well as in E-1, largely confirm the results obtained in the case study.

Regarding the generation of classes, the obtained results confirm that the set of the main BPMs enables the generation of a very complete CDM, because compared to nine real CDMs designed based on the same text specification, a very high average recall ($\overline{R}$=0.95) was achieved ($\overline{R}$=1.00 in the case study). The achieved average precision for the generation of classes is $\overline{P}$=0.62 ($\overline{P}$=0.65 in the case study), and the average effectiveness of the class generation process is $\overline{F}$=0.74 ($\overline{F}$=0.79 in the case study).

Regarding the generation of associations, the average completeness of the generated CDM comparing to the manually designed CDMs is $\overline{R}$=0.64 ($\overline{R}$=0.72 in the case study), with an average precision $\overline{P}$=0.22 ($\overline{P}$=0.33 in the case study), so the average effectiveness of the association generation process is $\overline{F}$=0.31 ($\overline{F}$=0.45 in the case study). The relatively low precision ($\overline{P}$=0.22) confirms that the generator generates a significant number of redundant associations (compared to the manually designed CDMs).

Based on the results, it can be concluded that the reference set of the main BPMs is a very good starting point for generating the target CDM, because the average completeness of the generated CDM is 95% for classes and 64% for associations, compared to the manually designed CDMs.

Finally, given the results obtained in E-1 and E-2, we can conclude that ”each” real set of the main BPMs, created by experienced modelers, will constitute a very good starting base for the automatic generation of the target CDM.

6.4. Experiment #3 (E-3)

Experiment design. An earlier experiment [20] with database practitioners as participants, showed that an automatically generated CDM provides a solid basis for designing the target CDM, because it speeds up the design process compared to designing from scratch. Therefore, E-3 was conducted, which aimed to assess how much the CDM, which is automatically generated based on the reference set of the main BPMs, is a good starting point for designing the target CDM, instead of designing the target CDM from scratch.

The participants in E-3 were mostly the students who also participated in E-2. The task in E-3 was to use the CDM, which was generated based on the reference set of the main BPMs of the Faculty Library, as a starting point for designing the target CDM of the Library. The participants were able to discard excessive concepts, correct incorrectly generated concepts and/or add missing concepts in the automatically generated CDM. Since all participants already had the appropriate domain knowledge acquired in E-2 and had already designed a CDM from scratch, in E-3 each participant individually completed the task, which resulted in 12 CDMs[33] of the Library that are designed based on the CDM that was automatically generated based on the reference set of the main BPMs. All resulting CDMs were analyzed by teachers and graded as adequate.[34] Afterwards, each manually designed CDM was compared with the initial automatically generated CDM, and the same metrics and measures were used in the evaluation as in the other experiments.

Results. The results of comparing the CDMs, which were manually designed based on the automatically generated CDM, with the CDM that was automatically generated based on the reference set of the main BPMs, are shown in Table 5. Each table row contains results for the corresponding manually designed CDM – the first column contains the ID of the manually designed CDM, while the other columns contain the corresponding metrics and measures for automatically generated classes and automatically generated associations, respectively. The N-labeled column contains a number of the corresponding

[33] Available at: *https://gitlab.com/m-lab-research/amadeos-exp-2020/-/tree/master/E-3/CDMs*

[34] Analysis and evaluation were conducted by the same teachers as in E-2.

concepts in the manually designed CDM. Figure 18 shows a comparison of R, P, and F for the classes and associations in the automatically generated CDMs (based on the reference set of the main BPMs) in comparison with the manually designed CDMs (based on the automatically generated CDM).

Table 5. Assessment of automatically generated CDM (based on reference set of main BPMs) in comparison with manually designed CDMs (based on automatically generated CDM)

CDM	Assessment of classes								Assessment of associations							
ID	N	N_g	N_c	N_m	N_w	R	P	F	N	N_g	N_c	N_m	N_w	R	P	F
01	15	20	13	2	7	0.87	0.65	0.74	22	55	14	8	41	0.64	0.25	0.36
02	10	20	10	0	10	1.00	0.50	0.67	14	55	9	5	46	0.64	0.16	0.26
03	13	20	13	0	7	1.00	0.65	0.79	34	55	31	3	24	0.91	0.56	0.70
04	17	20	17	0	3	1.00	0.85	0.92	25	55	22	3	33	0.88	0.40	0.55
05	15	20	15	0	5	1.00	0.75	0.86	36	55	36	0	19	1.00	0.65	0.79
06	18	20	18	0	2	1.00	0.90	0.95	36	55	34	2	21	0.94	0.62	0.75
07	17	20	17	0	3	1.00	0.85	0.92	24	55	22	2	33	0.92	0.40	0.56
08	12	20	10	2	10	0.83	0.50	0.63	12	55	8	4	47	0.67	0.15	0.24
09	9	20	9	0	11	1.00	0.45	0.62	11	55	11	0	44	1.00	0.20	0.33
10	18	20	18	0	2	1.00	0.90	0.95	26	55	17	9	38	0.65	0.31	0.42
11	18	20	18	0	2	1.00	0.90	0.95	33	55	24	9	31	0.73	0.44	0.55
12	10	20	10	0	10	1.00	0.50	0.67	12	55	8	4	47	0.67	0.15	0.24
Mean	**14.3**	**20**	**14**	**0.3**	**6**	**0.98**	**0.70**	**0.80**	**23.8**	**55**	**19.7**	**4.1**	**34.9**	**0.80**	**0.36**	**0.48**

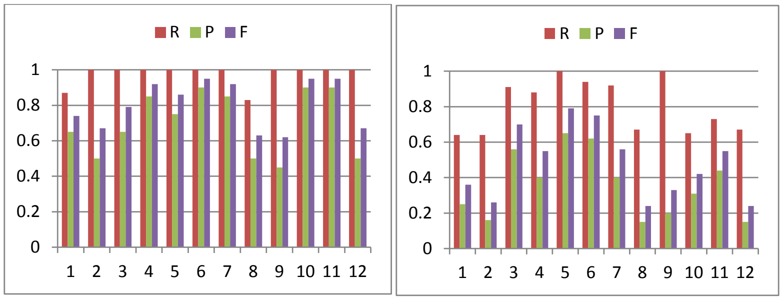


Fig. 18. Comparison of R, P, F for generated classes (left) and associations (right) in automatically generated CDMs (based on reference set of main BPMs) in comparison with manually designed CDMs (based on automatically generated CDM)

Discussion. The results obtained in E-3 confirm the previously obtained results in the experiment [20] with database professionals as participants. The results show that the CDM, which is automatically generated based on the set of the main BPMs, provides a good basis for manually designing the target CDM, rather than designing the target CDM from scratch.

Regarding the generation of classes, the obtained results confirm that the CDM, which is automatically generated based on the set of the main BPMs, enables the generation of a very complete CDM, because with 12 CDMs designed based on the automatically generated CDM, very high average recall ($\overline{R}$=0.98) is achieved ($\overline{R}$=1.00 in the case study). High average recall and high average precision $\overline{P}$=0.70 ($\overline{P}$=0.65 in the case study), resulted in high effectiveness ($\overline{F}$=0.80) of the class generation process ($\overline{F}$=0.79 in the case study).

Regarding the generation of associations, the average recall for the generated CDM in comparison to the manually designed CDMs is $\overline{R}$=0.80 ($\overline{R}$=0.72 in the case study), while the average precision is $\overline{P}$=0.36 ($\overline{P}$=0.33 in the case study). This means that the average effectiveness of generating associations is $\overline{F}$=0.48 ($\overline{F}$=0.45 in the case study). The results in E-3 also confirm that the generator generates a significant number of redundant associations (average precision for generating associations is $\overline{P}$=0.36).

The results obtained in E-3 are better than the results obtained in both the case study and in previous experiments, which was expected – the participants in E-3 had the automatically generated CDM as a starting point, so design was reduced mainly to discarding surplus from the initial CDM. Also, the designed models contain more concepts than CDMs that were manually designed from scratch – the average number of classes in CDMs that were manually designed from scratch in E-2 is 13, and in CDMs that were manually designed based on the automatically generated CDM is 14.3; while the average number of associations is 19 and 23.8, respectively.

6.5. Summative Evaluation Results

In order to achieve a better insight into the obtained results, we aggregated and compared the main results achieved in the case study and performed experiments. The summative evaluation results are presented in Table 6, where each row contains the corresponding R, P, and F values for the automatically generated classes and associations in a particular evaluation round. For the case study-based evaluation, the corresponding row contains values achieved by comparing the CDM derived from the reference set of the main BPMs with the reference CDM. For the experiments, each row contains average values achieved in the corresponding experiment. Finally, the bottom rows contain average values and the corresponding standard deviation (σ). The summative results are also shown in Fig. 19.

The summative overview of the results shows that the experiments largely confirmed the results obtained in the case study.

Regarding the generation of classes, the obtained results show that the implemented approach enables the generation of complete or almost complete CDM ($\overline{R}$=0.97, σ=0.03) with average precision $\overline{P}$=0.64 (σ=0.05), which gives the effectiveness of the process of generating classes $\overline{F}$=0.76 (σ=0.04).

Regarding the generation of associations, the obtained results show that the implemented approach enables the generation of an average of 68% of associations ($\overline{R}$=0.68, σ=0.10) with average precision $\overline{P}$=0.30 (σ=0.06), which gives the effectiveness of the process of generating associations $\overline{F}$=0.41 (σ=0.08).

The summative results show a relatively low precision ($\overline{P}$=0.30, σ=0.06) in generating associations, i.e. the tool generates a significant number of surplus associations in comparison to the target CDM.

Table 6. Summative evaluation results

Evaluation round	Classes			Associations		
	R	P	F	R	P	F
CS (CDM derived from reference set of main BPMs ↔ reference CDM)	1.00	0.65	0.79	0.72	0.33	0.45
E-1 (CDMs derived from students' sets of main BPMs ↔ reference CDM)	0.94	0.59	0.72	0.57	0.29	0.38
E-2 (CDM derived from reference set of main BPMs ↔ manually designed CDMs from scratch)	0.95	0.62	0.74	0.64	0.22	0.31
E-3 (CDM derived from reference set of main BPMs ↔ manually designed CDMs based on generated CDM)	0.98	0.70	0.80	0.80	0.36	0.48
Mean	**0.97**	**0.64**	**0.76**	**0.68**	**0.30**	**0.41**
Standard Deviation (σ)	**0.03**	**0.05**	**0.04**	**0.10**	**0.06**	**0.08**

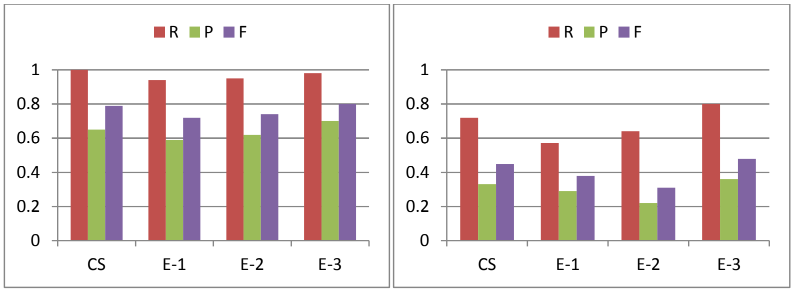


Fig. 19. Comparison of R, P, F for generated classes (left) and associations (right) achieved in case study-based evaluation and performed experiments

Comparison with results that could be obtained by applying other POM-based tools. For the same reference source set of BPMN models, the pre-existing version of the AMADEOS system is able to generate up to 46% of all classes, and up to 20% of all associations of the target reference CDM, since it is able to derive the CDM from only one single BPM from the source set.

Regarding the other tools taking BPMN models as a starting base for the CDM generation, we are only able to estimate[35] the completeness that could be achieved by their application to the same set of models. Since all these tools are able to derive the CDM from only one single BPM, like pre-existing AMADEOS, we conclude that they could not achieve higher completeness than pre-existing AMADEOS, since they implement a less complete set of transformation rules.

[35] As noted in Sect. 2, other tools are not publicly available, and we are not able to evaluate them.

6.6. Threats to Validity

In this section we consider possible threats to the validity of the experiments and derived conclusions, which may lie in the used models and the way the experiments were conducted.

Concerning the threats related to the source models, we have conducted E-1 in order to assess the approach effectiveness based on different real sets of BPMs, and not to draw conclusions based on a single set used in the case study. That is why we conducted the experiment with a large number of participants, which resulted in a relatively large number of sets for the same problem domain. In this way, we eliminated the threat that the initial (reference) set is predisposed to achieve maximum effectiveness.

Someone might consider that work in pairs of the participants in E-1 enabled better source BPMs. Since it is about novice modelers, it can be considered that we have just got relevant sets to be prepared by more experienced modelers and that the work of novice modelers in pairs is not a real threat. Working in pairs has additionally enabled us to analyze the appearance of accidental errors in the source sets, as well as possibilities for their elimination.

As for E-1, the only real threat may be that the source sets of BPMs were created in uncontrolled conditions to some extent (the participants created models as homework), so all sets might not be original enough. The performed analysis of the source sets found some plagiarisms and such duplicates were eliminated and not further used in the experiment. If one takes a look at the results achieved in E-1, it can be concluded that there are no identical results, which means there are no identical sets.

Regarding the threats related to the target models, we have conducted E-2 in order to assess to what extent the automatically generated CDM matches real CDMs manually designed for the same business system, and not to draw conclusions based on the one (reference) CDM used in the case study. Therefore, we conducted the experiment with a relatively large number of participants, which resulted in a larger number of target models. In this way, we eliminated the threat that the initial source set is predisposed to achieve maximum effectiveness in generating the CDM in comparison with one concrete or typical CDM in the given problem domain.

In addition to the manually designed CDMs from scratch, in E-3 we also used the CDMs that were manually designed on the basis of the automatically generated CDM. Although it was expected that more complete target models would be obtained at the time, some of the models with a very high recall for associations (R=1.00 for two models) can be suspected. The main reason that could have led to such high completeness is the relatively complex automatically generated CDM which was relatively difficult to edit in a web browser. This is confirmed by the fact that some participants in E-3 first exported the automatically generated CDM, then imported and edited it in some other tool. [36].

Although we used realistic sets of BPMs in the evaluation, it is nevertheless about relatively small sets that were created by small teams (pairs). Therefore, it is possible that, on the sets containing a bigger number of BPMs and created by more numerous teams, there could be a higher number of merging conflicts (e.g. synonyms) that could not be eliminated by simple techniques for composing partial CDMs. In such situations, we would have an additional decrease in precision and effectiveness.

[36] Note: See the CDMs designed in E-3.

The very important issue is related to the comparison of the automatically generated CDMs with the reference CDM, since the evaluation results are dependent on the comparison outcomes. Here we would like to emphasize that all the comparisons were performed manually by three evaluators, and all outcomes were achieved by their consensus.

6.7. Lessons Learned

This section presents some lessons we learned from the entire evaluation.

The results show that AMADEOS generates a significant number of surplus associations. This reduces the precision and effectiveness of the generation process and creates a feeling of dissatisfaction with designers (such an opinion was expressed by the majority of the participants in E-3) because they should manually eliminate surplus associations. Desirable tool improvement would be to enable the user/designer to influence the generation process by choosing which types of associations (s)he wants or does not want in the target model. Additional interviews with the E-3 participants showed that the tool does not provide appropriate comfort in editing complex models, so these AMADEOS' functionalities should be additionally improved in order to make the work easier for users.

The sets of BPMs in E-1 were prepared by pairs. The results show that inconsistencies happen in 44% of cases (12/27 sets were characterized by some kind of inconsistency, which resulted in redundant concepts in the automatically generated CDMs if the "Advanced Composition" option was applied during the generation). Potentially, the level of inconsistencies of the source sets could be higher in the case of larger project teams and larger sets of BPMs. Since AMADEOS currently allows the elimination of single typographical errors (*LD*=1), the possible AMADEOS' improvement could be increasing the level of user influence on elimination of inconsistencies so that the user would be able to set the value for *LD*.

6.8. Approach and Tool Limitations

In this section we discuss some limitations of the approach and implemented tool.

Although the approach enables the generation of a CDM structure with a fairly high level of completeness, the given set of transformation rules are not sufficient to enable automatic generation of the complete CDM. Some associations are still missing, as well as generalization relationships. The approach further enables just to automatically generate a CDM structure but does not enable automatic generation of attributes in the classes (except simple primary keys). These particular approach deficiencies were not in the primary focus of this article, but they constitute our main challenges and long-term research goals.

Although AMADEOS supports different source notations (BPMN and UML AD), and different serialization formats (XMI and XSD), currently all models in the entire source set must be represented by the same notation, and also serialized in the same format.

Although this particular research and conducted experiments did not face problems in merging partial CDMs, AMADEOS currently deploys very simple techniques that enable just to overcome inconsistencies related to different naming notations (case sensitivity) and accidental typing errors (Levenshtein distance) and therefore is not able to solve other types of merging conflicts which could appear.

Although the approach enables automatic generation of some kinds of associations that may be useful in some cases, the performed experiments showed that the tool generates a significant number of redundant associations that make it difficult to deal with the automatically generated CDM.

Although AMADEOS enables users to automatically generate the initial CDM, the forward database engineering based on the generated CDM is not possible, but users can only export the generated CDM and further use it in some other database design tool.

7. Conclusions

In this article we presented an approach to automatic CDM derivation from a set of BPMs, and the corresponding tool that implements the proposed approach. The approach proposes the incremental synthesis of the target model by iteratively composing the partial CDMs that are derived from the models contained in the source set. The implemented AMADEOS tool is the first online web-based tool that publicly enables automatic CDM derivation from a set of BPMs that may be represented by two different notations (BPMN or UML AD).

The approach effectiveness was evaluated in a case study and a series of experiments. As expected, the evaluation results confirmed that a set of BPMs enables the generation of a more complete CDM relative to the individual BPMs contained in the given set. The results show that the main BPMs constitute a sufficient basis to derive the target CDM since the auxiliary BPMs do not increase the recall achieved by deriving the CDM from the set of the main BPMs.

The summative results show that the implemented approach allows the generation of complete or almost complete CDM when it comes to classes (average completeness of generated models is 97%), which with an average precision of 64% gives average effectiveness of 76% of the class generation process. Regarding the generation of associations, the obtained results show that the implemented approach enables the generation of an average of 68% of the total number of associations, which with an average precision of 30%, gives average effectiveness of 41%. Analyzes show that the precision of generating associations is relatively low because the tool generates surplus associations (which are generally not incorrectly generated).

Future work will focus on further improving the approach and AMADEOS system, in line with the long-term research goals, the lessons learned in the experiments conducted, and the stated limitations. Future work will include: further identification of the semantic capacity of BPMs to increase the effectiveness of the CDM synthesis process, further tool improvements to provide users with greater comfort in working with complex models, and adding functionalities for the forward database engineering based on the generated CDM. Our intention is also to evaluate the approach with more complex real collections of BPMs.

Acknowledgments. The research is partially supported by Ministry for Scientific and Technological Development, Higher Education and Information Society of the Republic of Srpska, through Project no. 19.032/961-114/19: "*AMADEOS: Automatic derivation of conceptual database models from a collection of business process models*".

References

1. Aguilar, J.A., Garrigós, I., Mazón, J.N., Trujillo, J.: An MDA approach for goal-oriented requirement analysis in web engineering. Journal of Universal Computer Science 16(17), 2475–2494 (2010)
2. Alencar, F., Marín, B., Giachetti, G., Pastor, O., Pimentel, J.H.: From i* Requirements Models to Conceptual Models of a Model Driven Development Process. In: POEM 2009, LNBIP, vol. 39, pp. 99–114. Springer (2009)
3. Alencar, F., Pedroza, F., Castro, J., Amorim, R.: New mechanisms for the integration of organizational requirements and object oriented modeling. In: Proc. of WER 2003. pp. 109–123 (2003)
4. Alencar, F.M.R., Filho, G.A.C., Castro, J.F.: Support for Structuring Mechanism in the Integration of Organizational Requirements and Object Oriented Modeling. In: Proc. of WER 2002. pp. 147–161 (2002)
5. Alencar, F.M.R, Pedroza, F.P., Castro, J., Silva, C.T.L., Ramos, R.A.: XGOOD: A tool to automatize the mapping rules between i* framework and UML. In: Proc. of CIbSE 2006. pp. 125–138 (2006)
6. Ang, C.L., Khoo, L.P., Gay, R.K.L.: IDEF*: a comprehensive modelling methodology for the development of manufacturing enterprise systems. Int. Journal of Production Research 37(17), 3839–3858 (1999)
7. Banjac, D., Brdjanin, D., Banjac, G., Maric, S.: Evaluation of Automatically Generated Conceptual Database Model Based on Collaborative Business Process Model: Controlled Experiment. In: ICT Innovations 2016, AISC, vol. 665, pp. 134–145. Springer (2016)
8. Batini, C., Lenzerini, M., Navathe, S.: A comparative analysis of methodologies for database schema integration. ACM Comput. Surv. 18(4), 323–364 (1986)
9. Becker, L.B., Pereira, C.E., Dias, O.P., Teixeira, I.M., Teixeira, J.P.: MOSYS: A methodology for automatic object identification from system specification. In: Proc. of ISORC 2000. pp. 198–201. IEEE Computer Society (2000)
10. Bloomfield, T.: MDA, meta-modelling and model transformation: Introducing new technology into the defence industry. In: ECMDA-FA 2005, LNCS, vol. 3748, pp. 9–18. Springer (2005)
11. Boccalatte, A., Giglio, D., Paolucci, M.: ISYDES: the project of a tool aimed at information system development. In: Proc. of AIWORC 2000. pp. 293–298. IEEE (2000)
12. Brambilla, M., Cabot, J., Comai, S.: Automatic Generation of Workflow-Extended Domain Models. In: MoDELS 2007, LNCS, vol. 4735, pp. 375–389. Springer (2007)
13. Brambilla, M., Cabot, J., Comai, S.: Extending Conceptual Schemas with Business Process Information. Advances in Software Engineering, vol. 2010, Article ID 525121 (2010)
14. Brdjanin, D., Ilic, S., Banjac, G., Banjac, D., Maric, S.: Automatic derivation of conceptual database models from differently serialized business process models. Software and Systems Modeling 20(1), 89–115 (2021)
15. Brdjanin, D., Maric, S.: Towards the initial conceptual database model through the UML metamodel transformations. In: Proc. of Eurocon 2011. pp. 1–4. IEEE (2011)
16. Brdjanin, D., Maric, S.: An Approach to Automated Conceptual Database Design Based on the UML Activity Diagram. Computer Science and Information Systems 9(1), 249–283 (2012)
17. Brdjanin, D., Maric, S.: Model-driven Techniques for Data Model Synthesis. Electronics 17(2), 130–136 (2013)
18. Brdjanin, D., Maric, S., Gunjic, D.: ADBdesign: An approach to automated initial conceptual database design based on business activity diagrams. In: ADBIS 2010, LNCS, vol. 6295, pp. 117–131. Springer (2010)
19. Brdjanin, D., Banjac, D., Banjac, G., Maric, S.: Automated two-phase business model-driven synthesis of conceptual database models. Computer Science and Information Systems 16(2), 657–688 (2019)

20. Brdjanin, D., Banjac, G., Banjac, D., Maric, S.: An experiment in model-driven conceptual database design. Software & Systems Modeling 18(3), 1859–1883 (Jun 2019)
21. Brdjanin, D., Banjac, D., Banjac, G., Maric, S.: An Approach to Automated Two-phase Business Model-driven Synthesis of Data Models. In: Model and Data Engineering, LNCS, vol. 10563, pp. 57–70. Springer (2017)
22. Brdjanin, D., Banjac, G., Banjac, D., Maric, S.: Controlled Experiment in Business Model-driven Conceptual Database Design. In: Enterprise, Business-Process and Information Systems Modeling, LNBIP, vol. 287, pp. 289–304. Springer (2017)
23. Brdjanin, D., Banjac, G., Maric, S.: Automated Synthesis of Initial Conceptual Database Model Based on Collaborative Business Process Model. In: ICT Innovations 2014: World of Data, AISC, vol. 311, pp. 145–156. Springer (2015)
24. Brdjanin, D., Maric, S.: On Automated Generation of Associations in Conceptual Database Model. In: ER Workshops 2011, LNCS, vol. 6999, pp. 292–301. Springer (2011)
25. Brdjanin, D., Maric, S.: Towards the Automated Business Model-Driven Conceptual Database Design. In: Advances in Databases and Information Systems, AISC, vol. 186, pp. 31–43. Springer (2012)
26. Brdjanin, D., Vukotic, A., Banjac, G., Banjac, D., Maric, S.: Automatic Derivation of Conceptual Database Model from a Set of Business Process Models. In: 2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA). pp. 1–8. IEEE (2020)
27. Castro, J.F., Alencar, F.M.R., Filho, G.A.C., Mylopoulos, J.: Integrating organizational requirements and object oriented modeling. In: Proc. of ISRE 2001. pp. 146–153. IEEE (2001)
28. Conway, M.: How do committees invent? Datamation 14(4), 28–31 (1968)
29. Cruz, E., Machado, R., Santos, M.: Deriving a Data Model from a Set of Interrelated Business Process Models. In: Proc. of ICEIS 2015. pp. 49–59 (2015)
30. Cruz, E.F., Machado, R.J., Santos, M.Y.: From Business Process Modeling to Data Model: A systematic approach. In: Proc. of QUATIC 2012. pp. 205–210. IEEE (2012)
31. Cruz, E.F., Machado, R.J., Santos, M.Y.: On the Rim Between Business Processes and Software Systems. In: Cruz, A.M.R., Cruz, M.E.F. (eds.) New Perspectives on Information Systems Modeling and Design, pp. 170–196 (2019)
32. de la Vara, J.L.: Business process-based requirements specification and object-oriented conceptual modelling of information systems. PhD Thesis, Valencia Polytechnic Uni. (2011)
33. Dominguez, E., Pérez, B., Rubio, A., Zapata, M.A., Allué, A., López, A.: Generating persistence structures for the integration of data and control aspects in business process monitoring. In: Proc. of the 20th Int. Conf. on Enterprise Information Systems – Vol. 2: ICEIS. pp. 320–327. SciTePress (2018)
34. Drozdova, M., Kardos, M., Kurillova, Z., Bucko, B.: Transformation in Model Driven Architecture. In: Information Systems Architecture and Technology: Proceedings of 36th International Conference on Information Systems Architecture and Technology – ISAT 2015 – Part I. pp. 193–203. Springer, Cham (2016)
35. Drozdová, M., Mokryš, M., Kardoš, M., Kurillová, Z., Papán, J.: Change of Paradigm for Development of Software Support for eLearning. In: Proc. of ICETA 2012. pp. 81–84. IEEE (2012)
36. Dujlovic, I., Obradovic, N., Kelec, A., Brdjanin, D., Banjac, G., Banjac, D.: An Approach to Web-based Visualization of Automatically Generated Data Models. In: IEEE EUROCON 2019 – 18th International Conference on Smart Technologies. pp. 1–6. IEEE (2019)
37. España, S.: Methodological integration of communication analysis into a model-driven software development framework. PhD Thesis, Valencia Polytechnic Uni. (2011)
38. Essebaa, I., Chantit, S.: Toward an automatic approach to get PIM level from CIM level using QVT rules. In: 2016 11th International Conference on Intelligent Systems: Theories and Applications (SITA). pp. 1–6. Mohammedia (2016)
39. Fernandes, J.M., Lilius, J., Truscan, D.: Integration of DFDs into a UML-based model-driven engineering approach. Software and Systems Modeling 5(4), 403–428 (2006)

40. Fouad, A.: Embedding Requirements within the Model Driven Architecture. PhD Thesis, Bournemouth Uni. (2011)
41. Insfran, E., Pastor, O., Wieringa, R.: Requirements Engineering-Based Conceptual Modelling. Requirements Engineering 7(2), 61–72 (2002)
42. Insfran, E.: Requirements engineering approach for object-oriented conceptual modeling. PhD Thesis, Valencia Polytechnic Uni. (2003)
43. Jiang, L., Topaloglou, T., Borgida, A., Mylopoulos, J.: Goal-oriented conceptual database design. In: Proc. of RE '07. pp. 195–204. IEEE, Los Alamitos, USA (2007)
44. Jouault, F., Allilaire, F., Bezivin, J., Kurtev, I.: ATL: A model transformation tool. Science of Computer Programming 72(1-2), 31–39 (2008)
45. Khlif, W., Elleuch, N., Alotabi, E., Ben-Abdallah, H.: Designing BP-IS Aligned Models: An MDA-based Transformation Methodology. In: Proc. of the 13th Int. Conf. on Evaluation of Novel Approaches to Software Engineering – ENASE 2018. pp. 258–266 (2018)
46. Koch, N.: Transformation Techniques in the Model-Driven Development Process of UWE. In: Proc. of the Workshops at ICWE'06, Art. No. 3. ACM (2006)
47. Koch, N., Zhang, G., Escalona, M.J.: Model Transformations from Requirements to Web System Design. In: Proc. of ICWE'06. pp. 281–288. ACM (2006)
48. Koskinen, J., Peltonen, J., Selonen, P., Systa, T., Koskimies, K.: Model processing tools in UML. In: Proc. of ICSE 2001. pp. 819–820. IEEE Computer Society (2001)
49. Kriouile, A., Addamssiri, N., Gadi, T.: An MDA Method for Automatic Transformation of Models from CIM to PIM. American Journal of Software Engineering and Applications 4(1), 1–14 (2015)
50. Levenshtein, I.V.: Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady 10(8), 707–710 (1966)
51. Lingzhi, L., Ang, C.L., Gay, R.K.L.: Integration of Information Model (IDEF1) with Function Model (IDEF0) for CIM Information System Design. Expert Systems with Applications 10(3/4), 373–380 (1996)
52. Liu, D., Subramaniam, K., Far, B., Eberlein, A.: Automating Transition from Use-cases to Class Model. In: Proc. of CCECE 2003. pp. 831–834. IEEE (2003)
53. Martinez Rebollar, A.: Conceptual Schemas Generation from Organizational Models in an Automatic Software Production Process. PhD Thesis, Valencia Polytechnic Uni. (2008)
54. Nikiforova, O., Gusarovs, K., Gorbiks, O., Pavlova, N.: BrainTool: A tool for generation of the UML class diagrams. In: Proc. of ICSEA 2012. pp. 60–69. IARIA (2012)
55. Nikiforova, O., Gusarovs, K., Gorbiks, O., Pavlova, N.: Improvement of the Two-Hemisphere Model-Driven Approach for Generation of the UML Class Diagram. Applied Computer Systems 14(1), 19–30 (2013)
56. Nikiforova, O., Pavlova, N.: Application of BPMN instead of GRAPES for two-hemisphere model driven approach. In: ADBIS 2009 Workshops, LNCS, vol. 5968, pp. 185–192. Springer (2010)
57. OMG: MOF 2.0 Query/View/Transformation Specification, v1.0. OMG (2008)
58. OMG: Business Process Model and Notation (BPMN), v2.0. OMG (2011)
59. OMG: Unified Modeling Language (OMG UML), v2.5. OMG (2015)
60. Pottinger, R.A., Bernstein, P.A.: Merging models based on given correspondences. In: J.C. Freytag et al. (ed.) Procs. 2003 VLDB Conference, pp. 862–873. Morgan Kaufmann (2003)
61. Rhazali, Y., Hadi, Y., Chana, I., Lahmer, M., Rhattoy, A.: A Model Transformation in Model Driven Architecture from Business Model to Web Model. IAENG International Journal of Computer Science 45(1), 214–227 (2018)
62. Rodriguez, A., Fernandez-Medina, E., Piattini, M.: Analysis-Level Classes from Secure Business Processes Through Model Transformations. In: TrustBus 2007, LNCS, vol. 4657, pp. 104–114. Springer (2007)

63. Rodriguez, A., Garcia-Rodriguez de Guzman, I., Fernandez-Medina, E., Piattini, M.: Semi-formal transformation of secure business processes into analysis class and use case models: An MDA approach. Information and Software Technology 52(9), 945–971 (2010)
64. Rodriguez, A., Fernandez-Medina, E., Piattini, M.: Towards Obtaining Analysis-Level Class and Use Case Diagrams from Business Process Models. In: ER Workshops 2008, LNCS, vol. 5232, pp. 103–112. Springer (2008)
65. Rungworawut, W., Senivongse, T.: Using Ontology Search in the Design of Class Diagram from Business Process Model. PWASET 12, 165–170 (2006)
66. Santos, M.Y., Oliveira e Sá, J.: A Data Warehouse Model for Business Processes Data Analytics. Springer, Cham (2016)
67. Santos, M.Y., Machado, R.J.: On the Derivation of Class Diagrams from Use Cases and Logical Software Architectures. In: Proc. of ICSEA '10. pp. 107–113. IEEE (2010)
68. Selonen, P., Koskimies, K., Sakkinen, M.: Transformations Between UML Diagrams. Journal of Database Management 14(3), 37–55 (2003)
69. Sepúlveda, C., Cravero, A., Cares, C.: From Business Process to Data Model: A Systematic Mapping Study. IEEE Latin America Transactions 15(4), 729–736 (2017)
70. Silva, L.F., Leite, J.C.S.P.: Generating requirements views: A transformation-driven approach. Electronic Communications of the EASST 3, 1–14 (2006)
71. Srivastava, S.: Model Transformation Approach for a Goal Oriented Requirements Engineering based WebGRL to Design Models. International Journal of Soft Computing and Engineering (IJSCE) 3(6), 66–75 (2014)
72. Tan, H.B.K., Yang, Y., Blan, L.: Systematic Transformation of functional analysis model in Object Oriented design and Implementation. IEEE Transaction on Software Engineering 32(2), 111–135 (2006)
73. Truscan, D., Fernandes, J.M., Lilius, J.: Tool support for DFD-UML based transformation. In: Proc. of ECBS '04. pp. 378–387. IEEE (2004)
74. Winkler, W.E.: String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In: Proc. of the Section on Survey Research Methods. pp. 354–359. American Statistical Association (1990)
75. Wrycza, S.: The ISAC-driven transition between requirements analysis and ER conceptual modelling. Information Systems 15(6), 603–614 (1990)
76. Zhang, J., Feng, P., Wu, Z., Yu, D., Chen, K.: Activity based CIM modeling and transformation for business process systems. International Journal of Software Engineering and Knowledge Engineering 20(3), 289–309 (2010)

Drazen Brdjanin is an Associate Professor at the Faculty of Electrical Engineering, University of Banja Luka (Bosnia and Herzegovina), where he heads the M-lab Research Group. His research interests focus on information systems and software engineering. He has participated in several national and international R&D projects, and also authored a number of research papers and articles in the field of model-driven development.

Aleksandar Vukotic is a Master's Student at the Faculty of Electrical Engineering, University of Banja Luka (Bosnia and Herzegovina). He is a Senior Software Developer at RT-RK Auto and a member of the M-lab Research Group. His research interests include model-driven software development, databases, and UML. He has published a couple of research papers.

Danijela Banjac is a Senior Teaching Assistant and PhD student at the Faculty of Electrical Engineering, University of Banja Luka (Bosnia and Herzegovina). She is a member of the M-lab Research Group. Her research interests include model-driven software development, business process modeling, object-oriented information systems, and UML. She has published several research papers and articles.

Goran Banjac is a Senior Teaching Assistant and PhD student at the Faculty of Electrical Engineering, University of Banja Luka (Bosnia and Herzegovina). He is a member of the M-lab Research Group. His research interests include model-driven software development, business process modeling, databases, and UML. He has published several research papers and articles.

Slavko Maric is a Full Professor at the Faculty of Electrical Engineering, University of Banja Luka (Bosnia and Herzegovina). His current research interests include: information systems modeling, design and development, databases, eGovernment systems, service oriented architecture and parallel processing. He has published over 50 research papers and articles, and participated in a number of research and development projects.

Received: April 23, 2021; Accepted: December 14, 2021.