Methods and problems of analyzing process diagrams

S I Ryabukhin
Institute of mining - Federal Research Center of the Far Eastern Branch of the Russian Academy of Sciences, Khabarovsk, Russia
E-mail: poleem@mail.ru

Abstract. Process diagrams of are widely used to develop conceptual database models, in particular to establish the types of relationships between process elements. For some types of process diagrams, methods have been developed for converting them into sets of relationships. However, there are objective reasons that make it difficult to convert process diagrams into adequate and complete sets of relationships.

1. Introduction

The databases should have three levels of presentation, namely external, conceptual and internal [1]. The external level corresponds to the information picture of the domain, which is understandable for the users of the designed information systems. The conceptual layer is an external layer's interpretations and is referred to as the "conceptual database model" (CDM). At the conceptual level, a domain is described by a set of types of domain objects (concepts) and concept relationships. The inner layer corresponds to the logical model of the database (e.g., relational, network, or hierarchical) and the physical implementation of such a database in terms of the database management system. An interface must exist between the external and conceptual levels, as well as between the conceptual and internal levels [1]. This paper explores some of the methods and results of the implementation of the interface between the external and conceptual levels.

2. On the application of process diagrams for the purpose of synthesizing CDM

The currently existing methods for constructing the CDM, firstly, do not allow obtaining a set of concept relationships that is invariant to the external level of representation, and secondly, they allow the creation of alternative, including radically different sets of relationships. The reasons for this state of affairs are, inter alia, the use of intuitive and informal approaches to conceptual modeling, as well as the subjectivity of the quality criteria for CDM [2, 3]. In particular, in order to create CDM, the structure and content of process diagrams are often used. In this case, the procedure for constructing CDM, as a rule, is reduced to the intuitive identification of various types of relationships from the schemes of processes, for example, associative relationships of equivalent concepts; relationships of specialization; relationships of generalization (generalization); relationships of aggregation and relationships of composition. In order to identify the types of relationships from process diagrams, various heuristic methods are used. At the same time, the use of such methods contributes, but does not lead to the creation of complete, relevant and invariant sets of types of relationships [4].

As you know, the process domain model is a set of coordinated processes [5], which include the concepts involved in the execution of such processes [6]. Various notations are used to construct process
3. BPMN process flow diagram analysis methods

In this work, we investigate the methods and results of the synthesis of sets of concept relationships based on process diagrams, executed in BPMN notation. As a result of this synthesis, a set of relationships should be obtained between the main elements of BPMN diagrams: users, processes, tasks, events, objects, data stores.

Vara (et al.) [13] proposes to use the a posteriori class metamodel (UML) to identify relationships between elements of BPMN diagrams. Such a metamodel includes a set of classes and their relationships. Metamodel classes: Process; Case - process states and process status; ActivityType - functions and operations of processes; Activity - the states of functions and operations, as well as their status; Event; EventType - types of events; User - users responsible for starting processes. Metamodel classes have certain relationships: BelongsTo; Manages; PartOf; InstanceOf; Performs; Precedes. Based on the metamodel and by means of the CASE application, the corresponding special cases of relationships of the elements of specific process diagrams are established [14]. The modality and cardinality of the identified relationships are established exclusively by hand based on the analyst's knowledge of the domain.

Another intuitive way of converting BPMN schemas into CDM (namely, in a class model in UML) is described in Cruz [15]. The proposed method is based on a priori informal rules for interpreting BPMN schemas, in particular: 1) any data storage BPMN schema is an entity; 2) the role of the participant (i.e. "pool" in the BPMN diagram) is an entity; 3) the link between participant's role with the data warehouse is described by the relationship of association with cardinality 1:n; 4) the cyclical link between participant's role with the data warehouse, which generates many instances of such a link, is described by the relationships of association with cardinality n:m.

In contrast to the above intuitive methods, Brdjanin (et al.) [16, 17] propose a method based on algorithmic procedures for analyzing BPMN circuits. Typically, BPMN diagrams are created using visual editors (for example, Microsoft Visio, Bizagi Modeler, etc.). Such editors allow you to automatically generate a description of BPMN diagrams in XML. The method for converting BPMN schemas to CDM, proposed by Brdjanin (et al.), is based on the analysis of an XML listing and allows, firstly, to identify elements of BPMN schemas, secondly, to identify relationships between such elements, and thirdly, to establish the types of relationships between elements.

As you know, the description of BPMN schemas in XML language has a specific "serialized" form. For example, figure 1 shows a fragment of a diagram in BPMN notation, and figure 2 shows its description in XML.

![Figure 1](image-url) An example of a BPMN diagram.

Since BPMN notation is a standardized specification of processes [11], the identification of BPMN schema elements from its XML specification is achieved by comparing such a specification with the
corresponding standard XSD prototype [11]. In this case, the types of relationships are established based on the links between the elements of BPMN schemas and the context of such links.

It is known that the direct links between the basic elements of BPMN schemas are regulated. In particular, links between process actions cannot cross the boundaries of the performers’ areas of responsibility (i.e. pools). For example, actions $F_1$, $F_2$ of pool $P_1$ and actions $F_3$, $F_4$ of pool $P_2$ in figure 1. Correspondingly, in the description of a BPMN schema in XML language, each element of the schema, firstly, is presented in the context of the boundaries of processes or pools, and secondly, it is described in the context of its admissible relationships with other elements. For example, valid links are the link between $F_1$ and $F_2$, and the link between $O_1$ and $F_1$ in figure 1. Thus, parsing the XML listing allows you to find only those relationships that correspond to direct as well as valid relationships of BPMN schema elements.

```
<?xml version="1.0" encoding="utf-8"?>
<Pools>
  <Pool Id="P" Name="P" Process="Process">
    <Lanes>
      <Lane Id="P1" Name="P1" ParentPool="P"></Lane>
      <Lane Id="P2" Name="P2" ParentPool="P"></Lane>
    </Lanes>
  </Pool>
</Pools>
<WorkflowProcesses>
  <WorkflowProcess Id="P" Name="P">
    <ActivitySets>
      <ActivitySet Id="F4" Name="F4"></ActivitySet>
    </ActivitySets>
    <Activities>
      <Activity Id="E1" Name="E1"></Activity>
      <Activity Id="F1" Name="F1"> <Implementation> <Task /></Implementation> <Performers /></Activity>
      <Activity Id="F2" Name="F2"> <Implementation> <Task /></Implementation> <Performers /></Activity>
      <Activity Id="E2" Name="E2"> <Event> </Event> <Activity>
      <Activity Id="F3" Name="F3"> <Implementation> <Task /></Implementation> <Performers /></Activity>
    </Activities>
    <DataObjects>
      <DataObject Id="O1" Name="O1"></Object>
      <DataObject Id="O2" Name="O2"></Object>
      <DataObject Id="O3" Name="O3"></Object>
    </DataObjects>
    <Transitions>
      <Transition Id="E1-F1" From="E1" To="F1"> <Condition /></Transition>
    </Transitions>
    <DataAssociations>
      <DataAssociation Id="O1-F1" From="O1" To="F1"></DataAssociation>
    </DataAssociations>
  </WorkflowProcess>
</WorkflowProcesses>
</Package>
```

**Figure 2.** An example of a BPMN schema description by XML.
The BPMN schema analysis method proposed by Brdjanin (et al.) [16, 17] allows one to detect the following relationships: 1) “participant-participant” associations, for example, the relationship $R1$ and $R2$ in figure 3; 2) “participant-object” associations, for example, the relationship $R3$; 3) “object-object” associations, for example, the relationship $R4$; 4) associations “the state of the object during the execution of the process by the performer”, for example, the relationship $R5$ in figure 3. This method also allows us to identify the relationship of the “part-whole” type between certain elements of the diagram, for example, in the case of the relationship $R1$ in figure 3. In particular, the type of relationship $R1$ is determined based on the nesting of the corresponding tags in the structure of the XML document, for example, the Pool, Pools, Lanes, Lane tags in figure 2.

It is easy to see that the relationship between objects $O1$ and $O2$ in figure 3 is valid. However, it is not possible to establish the type of relationship between such objects. An exception is associative relationships between such objects. For example, the type of relationship $R4$ between objects $O2$ and $O3$ can also be identified exclusively as associative. At the same time, a different type of relationship may well take place between the objects $O2$ and $O3$. In particular, object $O2$ can be part of object $O3$, for example, be part of it if $O3$ is a set. In this case, to establish the type of relationship between $O2$ and $O3$, it will be necessary to perform a syntactic and semantic analysis of the name of the action connecting the objects $O2$ and $O3$, and to perform the procedure for analyzing and comparing all the links of the object $O2$ in the context of the entire BPMN scheme as a whole. Such a procedure, for example, can make it possible to detect a single (or, on the contrary, multiple) occurrence of the object $O2$ in various sets, and, thereby, create the possibility of a more accurate determination of the type of relationship. For example, in the first case, there will be a “part-whole” relationship, and in the second case, “part off”. It is obvious that such a procedure, in fact, should implement a mechanism for forming conclusions, for example, through deductive inference.

In figure 3, invalid links between BPMN diagram elements, as well as relationships that are not identified through the method described above, are marked with a dash.

Figure 3. An example of the relationships between elements of a BPMN diagram.

The method for identifying relationships between entities of conceptual data models based on BPMN diagrams was further developed and implemented in the form of CASE [18, 19]. At the same time, the
method of studying the content of BPMN diagrams by parsing their descriptions in XML leads to the appearance of incomplete sets of relationships between the elements of such diagrams.

4. Conclusion
As a rule, procedures for constructing conceptual database models are of a heuristic nature. At the same time, the disadvantages of the resulting conceptual models, incl. the incompleteness of the set of relationships is eliminated in the process of an expert assessment of their content. To obtain more complete and adequate sets of relationships, the analysis of process diagrams is used. We investigated the reasons for the incompleteness of sets of relations in cases when such sets are generated as a result of analyzing the description of BPMN process schemas in XML. It is shown that data structures in XML documents describing BPNM schemas allow establishing only a limited set of types of relations between elements of such schemas. It has been established that for a more accurate identification of types of relationships based on BPNM schemas, in addition to the apparatus for analyzing XML data structures, it is also required to use the apparatus for inference, based, for example, on the use of deductive logic [20].

References
[1] Interim Report: ANSI/X3/SPARC Study Group on Data Base Management Systems Bulletin of ACM SIGMOD 75-02-08 1975 7 ed T Steel I-140
[2] Moody D 2005 Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions Data and Knowledge Engineering 55(3) 243-76
[3] Kashif M, Samira C and Isabelle W 2009 Data Quality through Conceptual Model Quality - Reconciling Researchers and Practitioners through a Customizable Quality Model Proc. of the 2009 Int. Conf. on Information Quality 61-74
[4] Brdjanin D and Maric S 2013 Model-driven techniques for data model synthesis Electronics 17 130-6
[5] Optner S L 1969 System Analysis for Solving Business and Industrial Problems (Moscow: Soviet radio)
[6] Mark D A and McGowan K 1993 Methodology of Structural Analysis and Design (Moscow: MetaTechnology)
[7] FIPS Publication 183 released of IDEF0 Retrieved from: http://www.nist.gov/cgi-bin
[8] Edward Y 2005 Structured Programming and Structured Design as Art Forms Proc. of the National Computer Conf. 1 277
[9] Process Description Capture Method Report Retrieved from: http://www.staratel.com/iso/IDEF/IDEF3/idef3.pdf
[10] Mendling J 2008 Event-Driven Process Chains (EPC) In: Metrics for Process Models. Lecture Notes in Business Information Processing 6 (Berlin: Springer, Heidelberg) 17-57
[11] Business Process Model and Notation v.2.0.2 Retrieved from: http://www.omg.org/spec/BPMN/2.0.2
[12] Muller R J 2002 Databases and the UML Designing (Moscow: Publishing house "Lori") 420
[13] de la Vara J L, Fortuna M H, Sanchez J, Werner C M L and Borges M R S 2009 A Requirements Engineering Approach for Data Modeling of Process-aware Information Systems BIS 2009 ser. LNBI 21 J ed W Abramowicz (Berlin: Springer-Verlag) 133-44
[14] Brambilla M 2006 Generation of WebML Web Application Models from Business Process Specification Proc. of the Tool Presentation at 6th Int. Conf. on Web Engineering (ICWE ’06) 85-6
[15] Cruz E F, Machado R J and Santos M Y 2012 From Business Process Modeling to Data Model: A Systematic Approach Proc. of QUATIC (IEEE) 205-10
[16] Brdjanin D, Banjac G and Maric S 2014 Automated Synthesis of Initial Conceptual Database Model Based on Collaborative Business Process Model Proc. of the ICT Innovations (Springer) 145-56
[17] Brdjanin D, Banjac G, Banjac D and Maric S 2019 An experiment in model-driven conceptual database design Softw. Syst. Model 18(3) 1859-83
[18] Brdjanin D, Ilic S, Banjac G, Banjac D and Maric S 2021 Automatic derivation of conceptual database models from differently serialized business process models Software and Systems Modeling 20 89-115
[19] Brdjanin D and Ilic S 2019 Enterprise, Business-Process and Information Systems Modeling Proc. of 20th International Conference Rome Italy 352 ed I Reinhartz-Berger et al. (Cham: Springer) 344-58
[20] Ryabukhin S I 2020 Method for the synthesis of conceptual data models based on process domain models Control Systems and Information Technologies 4(82) 80-7