

# Comparing Process Models in the BPMN 2.0 XML Format

Sergey Ivanov <syuivanov@gmail.com>, Anna Kalenkova <akalenkova@hse.ru>

PAIS laboratory, National Research University Higher School of Economics, 125319, Kochnovskiy, 3, Moscow, Russia

**Abstract.** Comparing business process models is one of the most significant challenges for business and systems analysts. The complexity of the problem is explained by the fact there is a lack of tools that can be used for comparing business process models. Also there is no universally accepted standard for modeling them. EPC, YAWL, BPEL, XPDL and BPMN are only a small fraction of available notations that have found acceptance among developers. Every process modeling standard has its advantages and disadvantages, but almost all of them comprise an XML schema, which defines process serialization rules. Due to the fact that XML naturally represents hierarchical and reference structure of business process models, these models can be compared using their XML representations. In this paper we propose a generic comparison approach, which is applicable to XML representations of business process models. Using this approach we have developed a tool, which currently supports BPMN 2.0 [1] (one of the most popular business process modeling notations), but can be extended to support other business process modeling standards.

**Keywords:** business process modeling, business process comparison, BPMN 2.0 (Business Process Model and Notation), XML (eXtensible Markup Language), process mining.

**DOI:** 10.15514/ISPRAS-2015-27(3)-17

**For citation:** Ivanov Sergey, Kalenkova Anna. Comparing Process Models in the BPMN 2.0 XML Format. Trudy ISP RAN/Proc. ISP RAS, vol. 27, issue 3, 2015, pp. 255-266. DOI: 10.15514/ISPRAS-2015-27(3)-17.

## 1. Introduction

The availability of methods and tools capable to compare process models is crucial for business process analysts. Thus, for example, there can be a need to use comparing methods in order to find duplicates in repositories of process models. Finding duplicates is an essential task for those process analysts who wish to add a new process model to a process repository or even merge two repositories. The other obvious example is a comparison of a real and a reference process models. A

challenge here is to obtain a real process model. This problem can be solved in several ways, but the most effective known approach is a process model discovery. A new scientific discipline, process mining, can be applied for this purpose. The first type of process mining techniques, discovery, is used to construct models from event logs created by information systems [2].

Since the process model is discovered, we have a reference and a real process models. After that, we can move to the comparison of these two process models (Fig. 1).

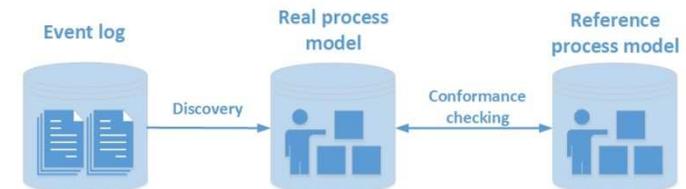


Fig.1. Conformance checking between two process models

The following approaches for comparing business process models are currently known: lexical matching, structural matching, and behavioral matching.

Lexical matching is based on the comparison of element labels. Labels comparison may include syntactic and semantic metrics for determining the accuracy between labels. Moreover, techniques for computing the string edit distance, such as the Hamming distance [3], the Levenshtein distance [4, 5], or the Damerau-Levenshtein distance [6] can be used. Each of these metrics is defined as a minimal number of operations needed to transform one string into the other using deletion, insertion, substitution of a single character, or transposition of two adjacent characters.

Also, a business process model can be transformed to a graph or a net. Therefore, process models can be compared as graphs by applying the graph-edit distance metric [7] (structural matching).

The behavioral matching is an approach, based on comparing the behavioral components of models. An algorithm based on causal footprints was suggested in [8]. A causal footprint provides a definition of a set of conditions on the order of activities that hold for the model.

Our approach is based on the fact that process models, which need to be compared, should be represented in XML format. Although this approach is described and implemented for process models represented in BPMN XML 2.0, it can be extended to compare process models defined using other XML formats due to the hierarchical nature of XML.

Note that we didn't find any special tool for comparison of two XML files in accordance with their XML schema.

## 2. Structure of XML schema

The structure of XML schema is a key factor for understanding the comparison algorithm proposed. In this section we will discuss the structure of XML schema by an example of the BPMN 2.0 XML schema format [9].

XML schema defines elements contained by an XML document and their types. Fig. 2 shows that BPMN 2.0 XML schema is represented by a list of elements descriptions and their complex (compound) and simple types.

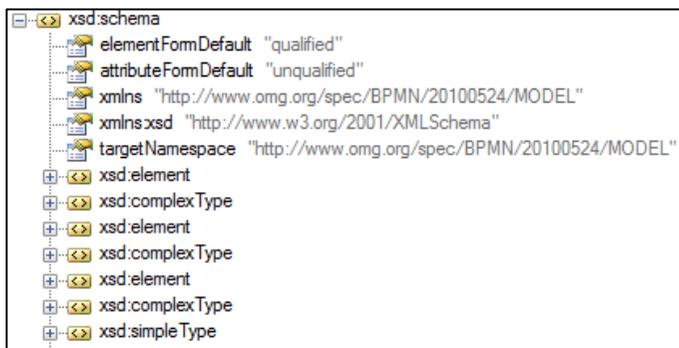


Fig.2. BPMN 2.0 XML schema

Let us consider a description of the element «subProcess» (Fig. 3).

```
<xsd:element name="subProcess"
            type="tSubProcess"
            substitutionGroup="flowElement"
/>
```

Fig.3. «subProcess» BPMN 2.0 XML element

Subprocesses in terms of BPMN represent multiple tasks that work together to achieve certain goals. The composite nature of subprocesses is reflected in a corresponding complex XML type (Fig. 4).

The type «tSubProcess» extends an abstract type «tActivity» with sets of lanes (containers used to logically organize activities within a subprocess), flow elements, which represent all the elements contained, and artifacts, which stand for the comments to subprocess elements. Attributes «minOccurs» and «maxOccurs», indicating the minimum and maximum number of occurrences of an element, show that each inner element can be presented zero or more times within a subprocess.

Thus, to compare subprocesses we need recursively compare all the contained elements.

```
<xsd:complexType name="tSubProcess">
  <xsd:complexContent>
    <xsd:extension base="tActivity">
      <xsd:sequence>
        <xsd:element
          ref="laneSet"
          minOccurs="0"
          maxOccurs="unbounded"/>
        <xsd:element
          ref="flowElement"
          minOccurs="0"
          maxOccurs="unbounded"/>
        <xsd:element
          ref="artifact"
          minOccurs="0"
          maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute
        name="triggeredByEvent"
        type="xsd:boolean"
        default="false"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

Fig.4. «subProcess» BPMN 2.0 XML element

The other element to be considered is a sequence flow (Fig. 5). Sequence flows are usually depicted as directed arcs and used to show the order, in which activities will be performed within a process. For each sequence flow identifiers of the source and the target nodes are specified using attributes of a special IDREF type. This should be taken into account during the comparison. Sequence flows and other connecting elements should be compared according to their source and target nodes, but not according to the identifiers of these nodes. In other words, two sequence flows coincide if their source and target nodes coincide, while nodes identifiers usually differ. This fact distinguishes our algorithm from other XML comparison algorithms, which don't consider element references.

Another important fact that should be taken into account is that XML schema contains abstract elements. Abstract elements are unavailable for end users, but used for inheritance. Their main purpose is to make language more extensible and allow adding new elements inheriting some parameters from their parents.

```
<xsd:element name="sequenceFlow"
            type="tSequenceFlow"
            substitutionGroup="flowElement"/>
<xsd:complexType name="tSequenceFlow">
  <xsd:complexContent>
    <xsd:extension base="tFlowElement">
      <xsd:sequence>
        <xsd:element name="conditionExpression"
                    type="tExpression"
                    minOccurs="0"
                    maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attribute name="sourceRef"
                    type="xsd:IDREF" use="required"/>
      <xsd:attribute name="targetRef"
                    type="xsd:IDREF" use="required"/>
      <xsd:attribute name="isImmediate"
                    type="xsd:boolean" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

Fig.5. «sequenceFlow» element and «tSequenceFlow» type

### 3. Comparison algorithm

Now let us turn to the description of the comparison algorithm. First we have to define the notion of equivalent elements. Two XML elements are equivalent if and only if:

- they have the same names;
- for each attribute of the first XML element there exists one and only attribute of the second XML element, which has the same name and the same value and vice versa; Note that for IDREF attributes corresponding linked XML elements must be equivalent;
- for each nested element of the first XML element there exists one and only one equivalent nested element of the second XML element and vice versa.

First let us impose restrictions on the structure of XML documents. Assume that elements with IDREF attributes don't have nested elements; assume also that there are no IDREF links to these elements from other XML elements. Note that these restrictions are justified for XML documents, containing information on hierarchical process structure (e.g. subprocesses) and sequence flows connecting arbitrary process nodes. The algorithm consists of three steps.

### 3.1 The first step

The first step includes generation of a set of elements that are directly nested in the root element «definitions» for each model (Fig. 6).

```
<definitions xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" id="1"
            targetNamespace="http://www.bizagi.com/definitions/1"
            xmlns="http://www.omg.org/spec/BPMN/20100524/MODEL">
```

Fig.6. XML element «definitions»

### 3.2 The second step

Now we have two sets of BPMN elements for two models at the first level. For each element from the first set we perform the following steps:

- select all elements with same name from the second set;
- if no elements were selected add an «error» message to the result of comparison;
- set the correspondence between the element from the first set and each selected element if:
  - they don't have nested elements and IDREF attributes, but they have the same sets of attributes with coinciding names and values;
  - there are correspondences between their nested elements and attributes, which can be obtained recursively using Step B.

If there are remaining elements from the second set with no corresponding elements add an «error» message to the result of comparison.

### 3.3 The third step

Consider all the elements with IDREF attributes for both models:

- set the correspondence relation between them if and only if linked XML elements are in correspondence relations and not-IDREF attributes coincide as well;
- remove redundant correspondences, which are not supported by IDREF attributes.

This algorithm assists in determining equivalent elements, but generally speaking there is no guarantee that equivalence relations will be constructed if multiple corresponding elements can be obtained for some element.

The algorithm was extended with an ability to specify relevant and non-relevant attributes.

The result of the comparison can consist of three types of messages, which describe main information about comparison:

- «error» - an error message;
- «warning» - an alert message;
- «info» - an information message.

A message takes an «error» status if the algorithm cannot find an equal element in another model. If for some reasons the algorithm cannot compare the non-relevant attributes of elements, a message should be added to a «warnings» list. A message should be added to an information list, if an element from the first model has more than one equal element from the other model.

#### 4. Implementation

After the structure of the XML schema is analyzed, the BPMN XML schema can be disassembled and transformed into an object-oriented model, which is implemented using some programming language.

We have developed our algorithms on the basis of ProM framework [10]. The ProM framework is a free open source product developed by the Eindhoven University of Technology. The algorithm for comparison two business process models in the BPMN 2.0 XML format was successfully implemented in ProM and can be used by business process analysts. Further, the main steps for applying a ProM plugin for comparing process models are shown.

#### 4.1 Importing resources

First, the following resources should be imported to ProM:

- Model1.bpmn - the first business process Model
- Model2.bpmn - the second business process Model
- Schema.xsd – BPMN XML schema

After importing, these resources are displayed in the «Workspace» tab (Fig. 7).



Fig.7. List of imported resources

#### 4.2 Selecting and applying plugin

After importing resources the user selects a necessary plugin from the plugin list in the «Actions» tab. «XML BPMN 2.0 Comparator» plugin should be selected in our case (Fig. 8).



Fig.8. Selection of the «XML BPMN 2.0 Comparator» plugin

#### 4.3 Analysis of the results

The results of the plugin's work are represented in an information window with the results which are divided into three groups: «error», «warning», «info» on the «Views» tab (Fig. 9).

The final report with results can be exported from the ProM in .txt and .html formats.

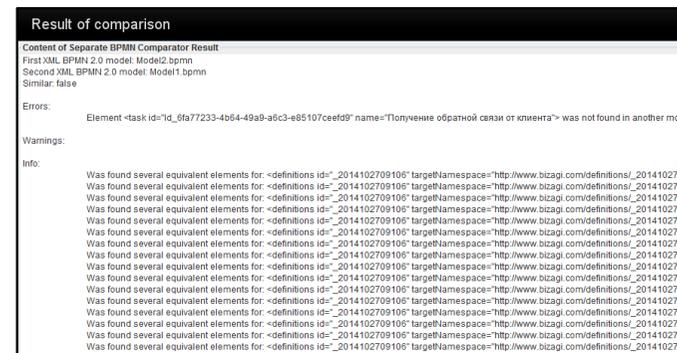


Fig.9. The result of the comparison of two models in the XML BPMN 2.0 format

#### 5. Example

Suppose we have a shopping process model (Fig. 10). This model includes start, end events and the following tasks: checking order information, saving an order to database, receiving of payment, delivering the goods. The delivery service is responsible for delivering an order. Delivering an order is a subprocess, which includes the following steps: collect order, test order, pack order, and deliver order. After a model is discovered from an event log, there is a need to compare the real

process model of e-shop (Fig. 10) with a reference process model (Fig. 11). These models should be imported to ProM framework and compared with «XML BPMN 2.0 Comparator» plugin.

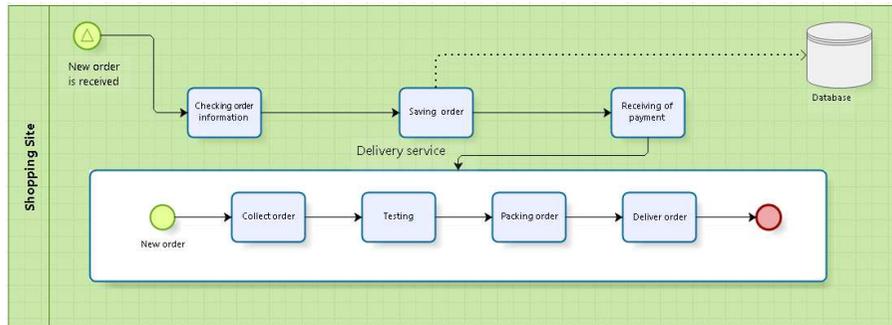


Fig.10. A real shopping process model

As a result plugin reported that an element with type «Task» and name «Testing» in the subprocess «Delivery service» was not found in a reference model. Also, a complete list of attributes, which were not found the document starting from the root element, was produced. According to the comparison results, analysts can find errors, modify and improve process of organization.

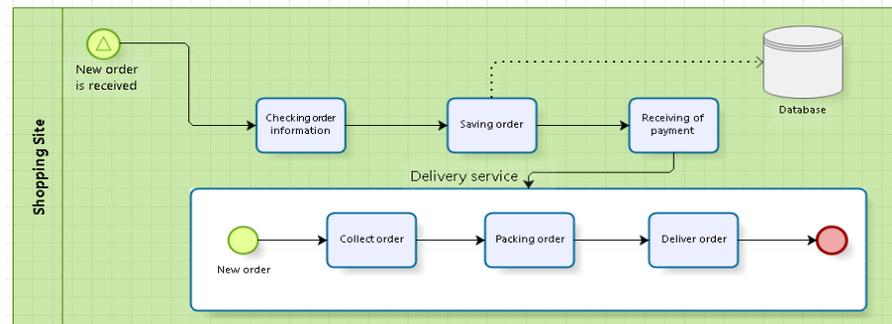


Fig.11. A reference shopping process model

## 6. Conclusion

Nowadays, system and business analysts face a problem of process models comparison due to the changes in processes occurring under influence of various factors. Therefore, there is a real demand for tools capable to compare process models.

This paper introduces a novel approach for process models comparison, which uses their XML representations.

We have proposed an algorithm that can be used to compare process models in XML format. This algorithm was described by the example of BPMN 2.0 XML format. The BPMN format was chosen as the most popular format for modeling business processes.

The results of the research were successfully implemented in the ProM framework and can be further used by business process analysts.

## Acknowledgment

This study was supported by Russian Fund for Basic Research (project 15-37-21103).

## References

- [1]. Stephen A. White. Introduction to BPMN [Online]. Available: [http://www.omg.org/bpmn/Documents/Introduction\\_to\\_BPMN.pdf](http://www.omg.org/bpmn/Documents/Introduction_to_BPMN.pdf)
- [2]. W. M. P. van der Aalst, Process Mining: Discovery, Conformance and Enhancement of Business Processes, Springer-Verlag, Berlin, Germany, 2011.
- [3]. D.Sanko and J. Kruskal, Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison, Addison-Wesley, 1983.
- [4]. V. Levenshtein, Binary codes capable of correcting spurious insertions and deletions of ones. Problems of Information Transmission, 1965, pp. 1-17.
- [5]. V. Levenshtein, Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady, pp. 10-707, 1966. Original in Russian in Doklady Akademii Nauk SSSR, 1965, pp. 163-848.
- [6]. F. Damerau. A technique for computer detection and correction of spelling errors. Comm. of the ACM, 1964, pp. 7-176.
- [7]. Xinbo Gao, Bing Xiao, Dacheng Tao, Xuelong Li, "A survey of graph edit distance" in Pattern Analysis and Applications, vol. 13, 2010, pp. 113-129.
- [8]. B.F. van Dongen, J. Mendling, and W.M.P. van der Aalst, "Structural Patterns for Soundness of Business Process Models" in EDOC 2006 – International Enterprise Distributed Object Computing Conference, Hong Kong, 2006, pp. 116-128.
- [9]. Object Management Group, "BPMN 2.0," [Online]. Available: <http://www.omg.org/spec/BPMN/2.0/>
- [10]. Process Mining Group, Eindhoven Technical University, "ProM 6," [Online]. Available: <http://www.promtools.org/>

## Сравнение моделей бизнес-процессов в формате BPMN 2.0 XML

Сергей Иванов <syuivanov@gmail.com>

Анна Каленкова <akalenkova@hse.ru>

НУЛ ПОИС, Национальный Исследовательский Университет Высшая Школа  
Экономики, 125319, Россия, г. Москва, пр. Кочновский, д. 3.

**Аннотация.** На сегодняшний день различным организациям приходится все чаще сталкиваться с моделированием своих бизнес-процессов для сокращения издержек и для обеспечения четкого понимания процессов, которые используются в организации. Но из-за изменения законодательства, внедрения инноваций и других факторов бизнес-процессы компании постоянно изменяются. В свою очередь системным и бизнес аналитикам, которые занимаются моделированием бизнес-процессов, нужен инструмент для сравнения моделей бизнес-процессов и определения их различий. Сложность решения данной проблемы объясняется недостатком инструментов, которые могут быть использованы для сравнения моделей бизнес-процессов. Также нет общепризнанного стандарта для моделирования. EPC, YAWL, BPEL, XPDL и BPMN только небольшая часть широко используемых нотаций, которые нашли признание среди разработчиков. Каждая нотация имеет свои преимущества и недостатки, но почти все из них описаны с помощью XML-схемы, которая определяет правила сериализации. В этой статье предложен общий подход к сравнению моделей бизнес-процессов, который опирается на XML представления моделей. Предложенный подход реализован в виде плагина для фреймворка ProM, который активно используется аналитиками и исследователями в рамках новой научной дисциплины process mining.

**Keywords:** business process modeling, business process comparison, BPMN 2.0 (Business Process Model and Notation), XML (eXtensible Markup Language), process mining.

**DOI:** 10.15514/ISPRAS-2015-27(3)-17

**Для цитирования:** Иванов Сергей, Каленкова Анна. Сравнение моделей бизнес-процессов в формате BPMN 2.0 XML. Труды ИСП РАН, том 27, вып. 3, 2015 г., стр. 255-266 (на английском языке). DOI: 10.15514/ISPRAS-2015-27(3)-17.

### Список литературы

- [1]. Stephen A. White. Introduction to BPMN [Online]. Available: [http://www.omg.org/bpmn/Documents/Introduction\\_to\\_BPMN.pdf](http://www.omg.org/bpmn/Documents/Introduction_to_BPMN.pdf)
- [2]. W. M. P. van der Aalst, Process Mining: Discovery, Conformance and Enhancement of Business Processes, Springer-Verlag, Berlin, Germany, 2011.

- [3]. D.Sanko and J. Kruskal, Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison, Addison-Wesley, 1983.
- [4]. V. Levenshtein, Binary codes capable of correcting spurious insertions and deletions of ones. Problems of Information Transmission, 1965, pp. 1-17.
- [5]. V. Levenshtein, Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady, pp. 10-707, 1966. Original in Russian in Doklady Akademii Nauk SSSR, 1965, pp. 163-848.
- [6]. F. Damerau. A technique for computer detection and correction of spelling errors. Comm. of the ACM, 1964, pp. 7-176.
- [7]. Xinbo Gao, Bing Xiao, Dacheng Tao, Xuelong Li, "A survey of graph edit distance" in Pattern Analysis and Applications, vol. 13, 2010, pp. 113-129.
- [8]. B.F. van Dongen, J. Mendling, and W.M.P. van der Aalst, "Structural Patterns for Soundness of Business Process Models" in EDOC 2006 – International Enterprise Distributed Object Computing Conference, Hong Kong, 2006, pp. 116-128.
- [9]. Object Management Group, "BPMN 2.0," [Online]. Available: <http://www.omg.org/spec/BPMN/2.0/>
- [10]. Process Mining Group, Eindhoven Technical University, "ProM 6," [Online]. Available: <http://www.promtools.org/>