

МЕТОДЫ ФОРМАЛИЗАЦИИ КОГНИТИВНОЙ ГРАФИКИ И ВИЗУАЛЬНЫХ МОДЕЛЕЙ С ИСПОЛЬЗОВАНИЕМ СХЕМ XML

А.И. Власов

vlasovai@bmstu.ru

Л.В. Журавлева

zhuravleva@bmstu.ru

В.В. Казаков

kazakov.vadim@yandex.ru

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

Проведен анализ методов формализации когнитивной графики и визуальных моделей с использованием перспективных форматов хранения данных. Дана характеристика основных визуальных проектных методик. Отмечено, что они характеризуются достаточной степенью изолированности друг от друга. Показано, что основной проблемой при комплексном использовании средств визуального моделирования является обеспечение связанности данных и знаний визуальных моделей различных уровней детализации. Проанализированы подходы к решению проблемы семантического разрыва, т. е. в случае выполнения условия однократности ввода характеристик рассматриваемых объектов, систем и процессов, следует осуществлять взаимную связанность данных из моделей, соответствующих разным уровням абстракции (экспертизы). Следует полагать, что основным недостатком подходов создания визуальных схем сложных систем является их фрагментация, изолированность, что ведет к локальной эффективности их использования. Предложены подходы к формализации визуальных моделей с использованием схем XML, что обеспечивает синхронность и взаимосвязь процессов разработки визуальных моделей разного уровня абстракции. На примере визуальной модели BPMN (Business, Process, Model and Notation) показаны принципы формального представления элементов визуальных моделей с использованием схем XML. Подробно проанализированы принципы взаимодействия слоев модели BPMN посредством гибкого описания на XML. Показано, что структура данных в BPMN имеет свою схему XML, содержащую все параметры клас-

Ключевые слова

Системный анализ, визуальные инструменты, схема XML, Business Process Model and Notation, когнитивная графика

са или элемента. Приведены примеры и методика применения в BPMN схемы XML и их дальнейшее обобщение для формального представления процессных моделей сложных систем	Поступила 03.03.2020 Принята 07.12.2020 © Автор(ы), 2021
---	--

Отдельные результаты получены при финансовой поддержке Минобрнауки России по проекту «Фундаментальные исследования методов цифровой трансформации компонентной базы микро- и наносистем» (проект № 0705-2020-0041)

Введение. Актуальность применения совокупности обоснованных, воспроизводимых и системных методов и средств визуального описания сложных систем и процессов в условиях цифровой трансформации промышленности определяет необходимость эффективного управления знаниево-ориентированной информацией по всем аспектам производственной деятельности. Каждая отдельно взятая производственная система представляет собой сложную систему, однако в настоящее время определения этого термина не существует. Отдельные подходы базируются на количественной оценке размерности системы. По мнению Г.Н. Поворова, сложной можно считать систему, включающую в себя 104–107 элементов, ультрасложной — систему из 107–1030 элементов, а суперсистемой — систему из 1030–10 200 элементов. Представленный подход имеет существенный недостаток — указанное определение сложности является не абсолютным, а относительным [1, 2]. С. Бир считает, что сложные системы — такие системы, которые можно описать на языке теоретико-вероятностных методов (мозг, экономика и т. д.) [3]. С учетом изложенного выше можно сделать вывод об определении сложной системы. Сложной системой называют систему, в модели которой существует нехватка данных для того, чтобы осуществлять эффективное управление ею [4]. Данное определение справедливо для классов систем, основным признаком которых является полнота данных для их моделирования и управления ими.

Таким образом, можно определить основной признак простоты системы — достаточное количество данных для осуществления управления системой. Систему полагают сложной, если результат управления, полученный с использованием модели, неожидан. Чтобы система принадлежала к разряду простых, необходимо собрать недостающую информацию о ней и добавить ее в модель [4–6]. Методы визуального моделирования — один из механизмов преобразования «сложная система → простая система». Такие методы обеспечивают иерархическую декомпозицию представленных объектов и процессов в исследуемой предметной области. Сове-

менный характер рассмотрения сложных систем определяет проблемы анализа происходящих в них процессов, трудоемкости построения формального описания, хранения и обработки знаний об объектах и процессах этих систем. В настоящее время в ходе поиска решения указанных проблем активно применяется визуальное моделирование и проектирование, которое служит главным инструментом генерации, хранения и обработки знаний о системе.

Принципы когнитивности, конвергенции (англ. *convergence* — схождение в одной точке) и инкапсуляции являются базой для построения графического (визуального) представления исследуемой предметной области, в том числе для описания сложных систем [7, 8]. Разработано множество различных инструментов для создания, изменения и редактирования визуальных моделей в цепочке модель–диаграмма–компонент [9–16]. Визуальное моделирование можно охарактеризовать как совокупность методов, применяющих метафоры визуализации. В свою очередь, они позволяют продемонстрировать объект с разных позиций. Кроме того, метафоры визуализации могут использоваться для эволюции и разработки моделирования [7, 8]. Сопоставление абстрактных или реальных объектов зрительно воспринимаемым образом представляет собой метафору визуализации. Фиксированные наборы метафор и правила построения с использованием визуальных моделей позволяют создать языки визуального моделирования.

Абстрактное визуальное моделирование обычно применяется на первоначальных этапах проектирования, оно позволяет представить исследуемые системы в упрощенном (обобщенном) виде (концептуальное моделирование). На этом этапе применяются формализованные методы, так как его основная цель — создание наиболее обобщенной и простой модели предметной области [14]. Такие модели получили название когнитивная графика. Общепринятого определения такого термина пока нет. Когнитивность (лат. *cognition* — познание, изучение, осознание) представляет собой желание и умение человека понимать и воспринимать информацию, а также ее обрабатывать (акт познания) [17]. Одни ученые понимают под когнитивной графикой совокупность методов, способов и приемов визуального представления исходных данных, которая позволяет либо сразу увидеть решение, либо получить подсказку для его нахождения. Однако это определение крайне спорно и не отражает познавательный характер. При этом под него попадают и различные варианты формализованных схем, для интерпретации которых необходимо владеть их условными графическими обозначениями. Другие ученые

под когнитивной графикой понимают набор методов и средств, способствующих познанию предметной области. Однако это определение не учитывает образность восприятия. В настоящей работе под когнитивной графикой будем понимать совокупность методов и средств образного познания предметной области, которое не требует для интерпретации образа знания специальных условных обозначений, т. е. можно констатировать, что элемент когнитивной графики — некоторый визуальный каллиграф, однозначно интерпретируемый в понятиях предметной области.

На следующем этапе проводится детализация высокой степени, когда визуальные модели разрабатываются с точки зрения структурно-функционального и операционного подходов. Здесь наиболее часто применяются модели IDEF [12]. Так, при моделировании информационных потоков выполняется построение моделей объектов и потоков данных. Кроме того, осуществляется построение информационной модели, структуры базы данных и базы знаний (при необходимости).

На заключительном этапе выполняется разработка объектно-ориентированных моделей системы с использованием процедур RUP. Именно на ее основе создается структура информационной системы, которая будет соответствовать реализуемым процессам [9, 11, 13, 15].

Подводя итоги краткого анализа уровней визуального моделирования, следует отметить, что большинство визуальных методик характеризуется достаточной степенью изолированности. Основной проблемой при комплексном использовании средств визуального моделирования является обеспечение миграции (связанности) данных и знаний моделей различных уровней экспертизы от одних инструментариев к другим. Указанная проблема получила название семантического разрыва. Суть проблемы состоит в том, что в случае выполнения условия однократности ввода характеристик рассматриваемых объектов, систем и процессов, следует осуществлять взаимную связанность данных из моделей, соответствующих разным уровням абстракции (экспертизы) [7]. Можно полагать, что основным недостатком представленных подходов создания визуальных схем сложных систем служит их фрагментация, изолированность применения визуального анализа, что ведет к локальной эффективности их использования. Легкость восприятия разработчиком визуальных схем (один рисунок заменяет тысячу слов, а одна схема тысячу рисунков) обуславливает необходимость использования эффективных способов описания, хранения и передачи информации о сложных системах между уровнями моделирования (экспертизы) в форме визуальных схем со встроенным внутренним универсальным метаописанием, единым для всех компонентов визуальной модели.

Цель работы — исследование подходов к формализации визуальных моделей с использованием схем XML, что должно обеспечить синхронность и взаимосвязь процесса разработки визуальных моделей разного уровня абстракции.

Суть предлагаемого метода визуального системного проектирования заключается в том, что на всех уровнях системного проекта элементы и диаграммы комплексной визуальной модели должны быть представлены единым унифицированным метаописанием XML, обеспечивающим инкапсуляцию базовых визуальных компонентов в единую замкнутую знаниево-ориентированную семантическую сеть. Это позволяет реализовать комплекс средств, обеспечивающих решение задач системного проектирования сложных систем, которые характеризуются применением визуальных схем на всех этапах проектирования. На основе обобщения применения различных форматов данных в работе предложен способ описания визуальных моделей сложных технических систем на основе метаязыка ViXML, который позволяет представлять в едином формате визуальные схемы на всех этапах исследования предметной области [18–21]. Результаты работы могут быть использованы при разработке перспективных средств CASE, обеспечивающих создание визуальных схем на всех этапах проектирования.

Материалы и методы решения задач, принятые допущения. Различные визуальные методы отличаются не только составом и характером моделей, которые разрабатываются в ходе проекта, но и подходами к их формализации. Как правило, каждая модель визуального проекта включает в себя текстовую и графическую нотации. На каждом этапе проектирования требования к составу, структуре и полноте графической и текстовых нотаций должны быть сформированы исходя из принципа «минимальной достаточности», так как это определяет скорость и трудоемкость разработки.

Немаловажный аспект — выбор эффективного способа (формата) представления информации, обеспечивающего сохранность информации о рассматриваемых системах и переносимость (экспорт/импорт) этой информации. Хранение информации в базе данных не обеспечивает переносимость в явном виде на физических носителях и требует знаний языков манипулирования данными (ЯМД) для извлечения информации. Бинарные файлы не могут быть прочитаны без соответствующего ПО или требуют знания специальных алгоритмов их записи/чтения. Текстовые форматы информации лишены подобных недостатков.

Проанализируем основные текстовые форматы хранения данных: CSV (англ. *Comma Separated Values* — значения, разделенные запя-

тыми), RTF (англ. *Rich Text Format* — формат обогащенного текста), XML (англ. *eXtensible Markup Language* — расширяемый язык разметки) [15].

Текстовый формат CSV используется для представления данных из таблиц. Любой строке файла соответствует одна единственная строка таблицы. Разделительный символ (*delimiter*) — запятая (,) или точка с запятой (;) — служит для разделения значений в отдельных колонках. В зависимости от установленных настроек определяется непосредственно символ разделителя. Так, в США символ разделителя — запятая, в России — точка с запятой, это связано с тем, что в России запятые применяются для дробных чисел, в США — точка. Можно встретить значения, содержащие так называемые зарезервированные символы, например запятая, точка с запятой, новая строка. Они обрамляются символом двойные кавычки (""); если в значении встречаются кавычки, то в файле их принято записывать в виде двух кавычек подряд.

Формат RTF (разработчик *Microsoft* и *Adobe*, 1982) работает на разных платформах, он проприетарный. Служит для хранения текстовых документов. Периодически спецификация у формата менялась. Документы в таком формате могут обрабатываться различными текстовыми редакторами.

Рекомендованный консорциумом W3C* язык разметки формата XML используется для хранения структурированных данных, а также для обмена данными между программами, XML представляет собой более простое подмножество языка SGML (*Standard Generalized Markup Language*). Результаты сравнения перечисленных форматов приведены в таблице.

Использование XML имеет больше преимуществ, чем RTF или CSV. Единственный недостаток — избыточный синтаксис, но это перевешивается преимуществами. XML является предпочтительным, так как обеспечивает представление как табличных (реляционные данные из базы данных или больших таблиц), так и псевдоструктурированных (web-страницы или полнотекстовые документы) данных. По сути, XML представляет собой совокупность стандартных синтаксических правил, которые позволяют описывать структуры документов с использованием тэгов разметки (дескрипторов, взятых в угловые скобки «<» и «>»). При этом теги изначально не определены и должны быть созданы разработчиком. Отсюда одно из главных преимуществ XML — он неограниченно расширяем. Это привело к созданию огромного числа подмно-

* <http://www.w3c.org>

жеств XML, например XHTML (семейство языков разметки веб-страниц*), XMCD (используется для описания данных программного пакета *Mathcad****) [15].

Сравнение форматов XML, RTF и CSV

Критерий	XML	RTF	CSV
Не зависит от платформы	Да	Да	Да
Имеет реализации парсеров для всех современных языков программирования	Да	Нет	Нет
Подходит для описания любых типов данных	Да, сетевые модели описываются с трудом	Нет, только текст	Нет, только таблицы
Открыт, т. е. не защищен патентами и может внедряться без дополнительных финансовых отчислений третьим лицам	Да	Нет	Да
Самодокументируемый формат	Да, описывает структуру, имена и значения полей	Нет	Нет
Избыточный синтаксис	Да, можно описать одну структуру несколькими равноправными способами	Нет	Нет
Расширяемость	Да, позволяет определять и использовать собственные словари	Нет	Нет

Как подмножество XML разработан и язык описания знаний о сложных технических системах ViXML, который обеспечивает гибкие возможности при настройке уровня формализма анализируемых объектов и процессов. Преимуществом минимально формализованного подхода является значительное сокращение времени проектирования за счет отсутствия временных затрат на разработку дополнительной документа-

* <http://www.w3.org/TR/xhtml11>

** <http://www.ptc.com/products/mathcad>

ции. Недостатки минимально формализованного подхода — риски, связанные с возможностью возникновения необходимости передачи проекта другой проектной команде, сложности в поддержке и переработке разработанной системы, высокая вероятность возникновения несогласованностей и прочих критических ошибок при проектировании. Преимущества высокоформализованного подхода — согласованность принятия решений по разработке, минимизация ошибок проектирования, простота обслуживания объекта разработки. Недостаток — огромные затраты сил и времени на разработку документации. Проанализируем особенности формализации визуальной модели BPMN инструментами XML.

Результаты. С позиции технической реализации структуры модели BPMN (версия 2.0) базой для ее построения выступает принцип расширения вышестоящих слоев главного ряда простых элементов, установленных в спецификации как элементы ядра (Core Elements) [23, 24]. Используя набор базовых конструкций модели [23], разделение на слои проводится для возможности введения дополнительных элементов спецификации, применяющихся для расширения уже существующих конструкций или добавления новых. Пути разрешения ситуации предопределяют разделение на слои. Такая модель, как XML Schema, подстраивается к другим моделям (структурная модель с импортом и механизмом принятия решений). Этот механизм позволяет сокращать неточности в процессе определения обозначений (терминов) для элементов, находящихся во внешних слоях.

Структура ядра и слоев BPMN показана на рис. 1. Подход заключается в использовании таких конструкций для расширения, которые достаточно формализованы. Еще одно преимущество разделения — возмож-

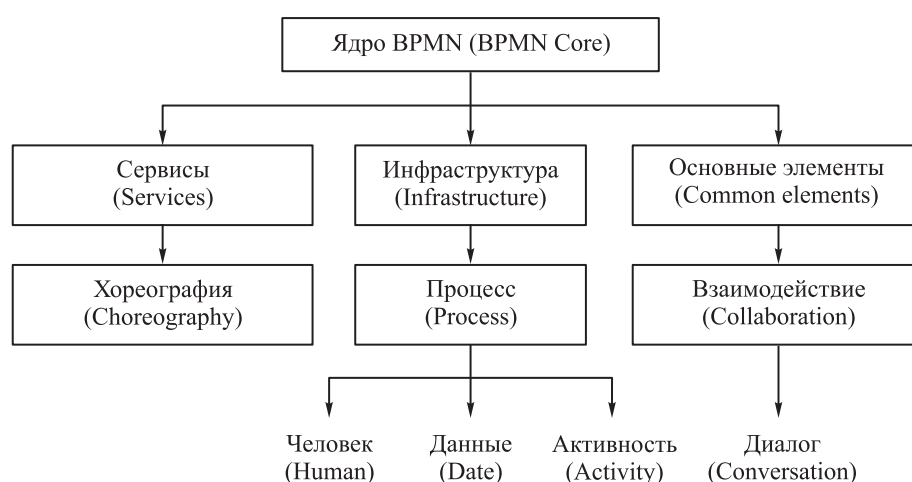


Рис. 1. Структура ядра и слоев BPMN

ность реализации совместимости слоев. Применяя это, сторонние разработчики инструментов моделирования могут использовать различные уровни совместимости. Исходя из этого, разработчики получают возможность определять свои слои для обеспечения поддержки всевозможных вертикальных областей для нужд целевой аудитории. Кроме того, обратная совместимость, или совместимость конкретной спецификации с чем-либо, в процессе разделения на слои не задействуется. Следует отметить, что суть разделения на слои заключается в возможности дать дополнительное определение знакомому термину.

В модели BPMN слои структуры располагаются на самой высокой позиции, тем самым они обозначают расширение слоев, которые находятся ниже в иерархии. Такая структура состоит из ядра (англ. *Core* или *Kernel*). Ядро включает в себя значимые элементы BPMN, которые используются при построении диаграмм. Это элементы: процесс, хореография и взаимодействие. При этом ядро удовлетворяет требованиям по простоте, компактности, расширяемости и учитывает поведенческие аспекты.

Ядро состоит из трех пакетов (рис. 2) [24]:

- 1) Foundation включает в себя базовые конструкции, используемые для моделирования диаграмм BPMN;
- 2) Service содержит базовые конструкции, используемые для создания моделей сервисов и интерфейсов;
- 3) Common состоит из классов, которые являются одинаковыми для слоев процесса, хореографии и взаимодействия.

Пакет Infrastructure BPMN включает в себя два элемента, которые нашли применение для создания моделей с абстрактным синтаксисом, а также построения диаграмм. Класс Definitions (входит в пакет Infrastructure) — объект, который является крайним для остальных элементов BPMN. С использованием принадлежащих к данному классу элементов можно выделить зону видимости и пространство имен таких элементов, которые включают в себя элементы класса. Для реализации обмена файлами BPMN необходимо применение одного или нескольких элементов Definitions (рис. 3).

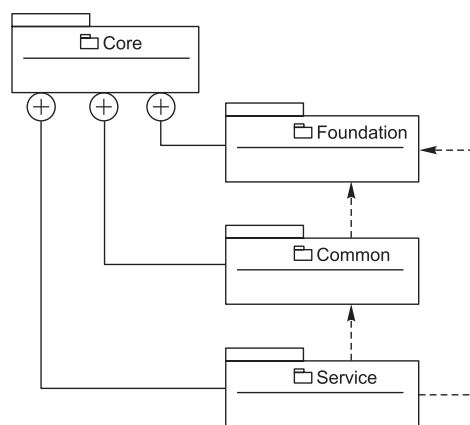


Рис. 2. Схема классов ядра BPMN

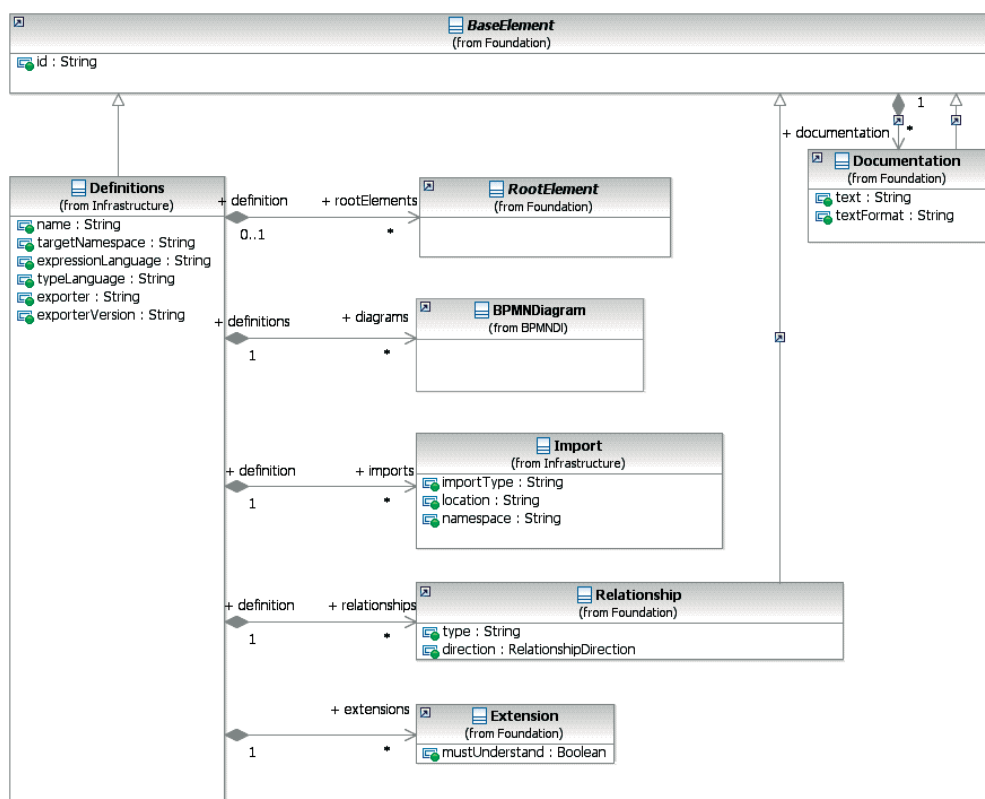


Рис. 3. Схема классов элемента Definitions [30]

Атрибуты и ассоциации элемента BaseElement имеют возможность наследовать элемент под названием Definitions (см. рис. 3). Внешним элементом может быть не только элемент BPMN, но и любой другой. Если присутствует ссылка на него, то применяется класс Import. Видимость элементов Import обязательна.

Шаблон XML элемента Definitions

```

<xsd:element name="definitions" type="tDefinitions"/>
<xsd:complexType name="tDefinitions">
  <xsd:sequence>
    <xsd:element ref="import" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="extension" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="rootElement" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="bpmndi:BPMNDiagram" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element ref="relationship" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" use="optional"/>

```

```
<xsd:attribute name="targetNamespace" type="xsd:anyURI" use="required"/>
<xsd:attribute name="expressionLanguage" type="xsd:anyURI" use="optional"
default="http://www.w3.org/1999/XPath"/>
<xsd:attribute name="typeLanguage" type="xsd:anyURI" use="optional"
default="http://www.w3.org/2001/XMLSchema"/>
<xsd:anyAttribute name="exporter" type="xsd:ID"/>
<xsd:anyAttribute name="exporterVersion" type="xsd:ID"/>
<xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>
```

Схема XML элемента Import

```
<xsd:element name="import" type="tImport"/>
<xsd:complexType name="tImport">
<xsd:attribute name="namespace" type="xsd:anyURI" use="required"/>
<xsd:attribute name="location" type="xsd:string" use="required"/>
<xsd:attribute name="importType" type="xsd:anyURI" use="required"/>
</xsd:complexType>
```

Пакет Foundation состоит из двух классов, каждый из которых распределен между остальными пакетами ядра модели с абстрактным синтаксисом.

Для множества остальных элементов BPMN BaseElement является абстрактным суперклассом. Он позволяет другим элементам наследовать значения его атрибутов id, а также документации.

Возможность других элементов BPMN, которые могут наследовать атрибуты или ассоциации элемента BaseElement, обладать одним или двумя текстовыми пояснениями, возникает, когда применяется элемент Documentation. Элемент Documentation может наследовать атрибуты и ассоциации элемента BaseElement (см. листинг схемы XML элемента Import). Схема XML элемента Documentation, а также сведения о дополнительных атрибутах этого элемента приведены ниже.

Схема XML элемента Documentation

```
<documentation>
  An example of how the documentation text is entered.
</documentation>
```

ComplexType tDocumentation в схеме BPMN не включает в себя каких-либо элементов или текстовых атрибутов. Чтобы решить эту проблему, непосредственно текст данного документа содержится в теле указанного элемента.

Следует отметить свойство метамодели BPMN — способность расширяться. Пользователи получили возможность расширять определен-

ные метамоделю при условии их совместимости с BPMN. Для расширения используется несколько определенных элементов. Такие элементы дают возможность пользователям применять не только типичные для BPMN элементы, но и некоторые другие.

Возможность расширения позволяет пользователям легко идентифицировать стандартные элементы. Кроме того, элементы могут заменять друг друга, а в процессе замены есть риск потери только дополнительных атрибутов и элементов.

Класс `Extension`, применяемый в создании моделей BPMN, обычно включает в себя четыре элемента:

- 1) `Extension`;
- 2) `ExtensionDefinition`;
- 3) `ExtensionAttributeDefinition`;
- 4) `ExtensionAttributeValue`.

`ExtensionDefinition` и `ExtensionAttributeDefinition` — главные элементы класса `Extension`. Число стандартных элементов BPMN было увеличено за счет добавления нового перечня атрибутов. В него включены сведения о названии, типе каждого нового атрибута. Такая информация позволяет пользователю BPMN выполнять интеграцию любой метамоделю в метамоделю BPMN. Кроме того, пользователь может применять уже используемые ранее элементы модели несколько раз, без ограничений.

Следует отметить важную особенность элемента `ExtensionDefinition` — возможность его создания независимо от какого-либо другого элемента или определения BPMN. Этот элемент должен иметь ассоциацию с элементом `Extension`, который применяется для создания связи элемента `ExtensionDefinition` с другими элементами, или определениями, модели BPMN, только тогда его можно применять в качестве элемента модели BPMN (элемента `Definitions`). К каждому элементу BPMN, который состоит в подклассе `BaseElement` этого BPMN, можно добавить дополнительные атрибуты.

Графический язык моделирования бизнес-процессов BPMN имеет ассоциацию с элементом `ExtensionDefinition`. Он обозначен среди элементов `Definitions` (определений) модели BPMN. Каждый дополнительный элемент BPMN включает в себя текущее значение атрибута расширения. Это значение содержит значения, имеющие тип `Element`, также оно определяется с использованием элемента `ExtensionAttributeValue`.

Расширение схемы XML

```
<xsd:element name="extension" type="tExtension"/>
<xsd:complexType name="tExtension">
```

```

<xsd:sequence>
<xsd:element ref="documentation" minOccurs="0" maxOccurs="unbounded"/>
</xsd:sequence>
<xsd:attribute name="definition" type="xsd:QName"/>
<xsd:attribute name="mustUnderstand" type="xsd:boolean" use="optional"/>
</xsd:complexType>

```

Существует возможность создания связи между элементом ExtensionDefinition, его атрибутами и определением модели BPMN посредством импорта в требующуюся модель BPMN с помощью элемента Extension.

Класс ExtensionDefinition используется для того, чтобы сгруппировать «расширенные» атрибуты, дать им определение. Однако необходимо учитывать, что такой тип невозможно применить при наличии схем XML ввиду того, что сложные типы (Complex Types) XSD соответствуют этому правилу.

Пример схемы XML

```

<xsd:schema ...>
...
<xsd:element name="task" type="tTask"/>
<xsd:complexType name="tTask">
<xsd:complexContent>
<xsd:extension base="tActivity"/>
</xsd:complexContent>
</xsd:complexType>
</xsd:schema>

```

С использованием элемента ExtensionAttributeDefinition можно добавить новые атрибуты. Здесь также требуется учесть, что такой тип неприменим, если была осуществлена замена схемы XML, так как механизмы XSD, обеспечивающие AnyAttribute или Any, соответствуют этому правилу.

Элемент ExtensionAttributeValue включает в себя значение атрибута. Такой тип нельзя применять при замене схемы XML вследствие того, что механизмы XSD, обеспечивающие AnyAttribute или Any, соответствуют этому правилу.

Пример расширения схемы XML

```

<xsd:schema ...>
...
<xsd:group name="dataRequirements">

```

```

<xsd:sequence>
  <xsd:element ref="dataInput" minOccurs="0" maxOccurs="unbounded" />
  <xsd:element ref="dataOutput" minOccurs="0" maxOccurs="unbounded" />
  <xsd:element ref="inputSet" minOccurs="0" maxOccurs="unbounded" />
  <xsd:element ref="outputSet" minOccurs="0" maxOccurs="unbounded" />
</xsd:sequence>
</xsd:group>
...
</xsd:schema>

```

Можно сформулировать цель проведенного анализа — обобщение основных элементов, применяющихся для создания синтаксически корректных моделей процессов с широкой семантикой. Модель такого типа может применяться при описании процессов (Processes), хореографий (Choreographies) и исполнительских действий всевозможных уровней абстракции. Как уже было отмечено, для этой спецификации расширением данных в BPMN можно решить задачу разнообразия таких данных, дополнения моделей для их детализации.

Модели таких процессов участвуют в довольно неоднозначных бизнес-процессах и процессах развития систем, они не представлены обособленно. Описание элементов, позволяющих осуществить интеграцию артефактов BPMN в процесс развития посредством уточнения ненавязчивой модели идентичности/взаимоотношений между артефактами (Artifacts) BPMN и элементами некоторой другой адресуемой модели в исследуемой предметной области, приведено ниже.

Экземпляр примера XML (XML instance)

```

<bpmn:definitions id="ID_1" ...>
...
  <bpmn:extension mustUnderstand="true"
definition="bpmn:dataRequirements"/>
...
  <bpmn:task name="Retrieve Customer Record" id="ID_2">
    <bpmn:dataInput name="Order Input" id="ID_3">
      <bpmn:typeDefinition typeRef="bo:Order" id="ID_4"/>
    </bpmn:dataInput>
    <bpmn:dataOutput name="Customer Record Output" id="ID_5">
      <bpmn:typeDefinition typeRef="bo:CustomerRecord" id="ID_6"/>
    </bpmn:dataOutput>
    <bpmn:inputSet name="Inputs" id="ID_7" dataInputRefs="ID_3"/>
    <bpmn:outputSet name="Outputs" id="ID_8" dataOutputRefs="ID_5"/>

```



```
</bpmn:task>
...
</bpmn:definitions>
```

Существующая модель Идентификатор/Ссылка сокращена для создания типизированных отношений, которые принадлежат к разным категориям. В результате образуются свободные взаимодействия между артефактами (принадлежащие и не принадлежащие к модели BPMN). Для включения модели BPMN в более приоритетные системные процессы развития и расширения моделей необходимо применять тип взаимоотношения (relationship type). Этот тип можно соотнести с элементами артефактов (принадлежащие и не принадлежащие к модели BPMN).

Подобные расширения позволяют создавать разнообразными методами взаимоотношения происхождения (derivation) и определения (definition) между артефактами UML и артефактами BPMN. По спецификации BPMN элемент диаграммы UML может быть соотнесен с элементом процесса без влияния непосредственно на артефакты. Это способствует осуществлению интеграции моделей, которые имеют отношения к данным взаимоотношениям.

Внешняя спецификация взаимоотношений расширения между артефактами BPMN и другими моделями классификации взаимоотношений (или внешними моделями) нашла применение при создании моделей. Внешние модели применяют для создания взаимосвязи между элементами BPMN, а также другими структурными и неструктурными определениями метаданных путем пересечения отношений, которые показаны во внешнем определении [25–29]. Модель UML для указанной спецификации идентична гибкой и несложной модели. В такой модели подобные взаимоотношения создаются посредством задания ссылки на соответствующие объекты, принадлежащие к определенному пространству имен. Например, есть возможность реализовывать взаимоотношения, возникающие между различными артефактами. Эти взаимоотношения дают возможность создавать внешние комментарии, которые применяются для классификаций, отслеживания, деривации и т. д. Такие взаимодействия легко прослеживаются между артефактом Visio™ и артефактом BPMN.

Схема XML реконструкции

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions targetNamespace=""
typeLanguage="" id="a123" expressionLanguage=""
xsi:schemaLocation="http://www.omg.org/spec/BPMN/20100524/MODEL Core-
Common.xsd"
```

```

xmlns="http://www.omg.org/spec/BPMN/20100524/MODEL"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:src="http://www.example.org/Processes/Old"
xmlns:tgt="http://www.example.org/Processes/New">
<import importType=http://office.microsoft.com/visio
location="OrderConfirmationProcess.vsd"
namespace="http://www.example.org/Processes/Old"/>
<import importType="http://www.omg.org/spec/BPMN/20100524/MODEL"
location="OrderConfirmationProcess.xml"
namespace="http://www.example.org/Processes/New"/>
<relationship type="reengineered" id="a234" direction="both">
<documentation>An as-is and to-be relationship. The as-is model is expressed as
a Visio diagram. The re-engineered process has been split in two and is captured in
BPMN 2.0 format. </documentation>
<source ref="src:OrderConfirmation"/>
<target ref="tgt:OrderConfirmation_PartI"/> <target
ref="tgt:OrderConfirmation_PartII"/>
</relationship> </definitions>

```

RootElement принадлежит к абстрактному суперклассу для всех существующих элементов BPMN, которые содержатся в элементах Definitions. Любой элемент имеет собственный цикл жизни. Следует отметить, что данные элементы невозможно удалить, даже когда другие подлежат удалению. Взаимодействие, процесс и хореография — образцы корневых элементов (RootElements). Корневые элементы (RootElements) используют ссылки на большое число других элементов в зависимости от возможностей их использования. Часть элементов RootElements также может включаться в другие элементы, кроме Definitions, чтобы не допустить сохранения побочных явлений не зависящих друг от друга жизненных циклов. Например, элемент EventDefinition содержится в Процессе (Process), так как применяется исключительно в нем. В этой ситуации элемент EventDefinition попадает в зависимость от жизненного цикла Процесса (Process). Элемент RootElement имеет возможность наследовать атрибуты и ассоциации элемента BaseElement, но другие атрибуты и ассоциации для него определить нельзя.

Схема XML для элемента RootElement

```

<xsd:element name="rootElement" type="tRootElement"/>
<xsd:complexType name="tRootElement" abstract="true">
<xsd:complexContent>
<xsd:extension base="tBaseElement"/>
</xsd:complexContent>
</xsd:complexType >

```

Схема XML для элемента BaseElement

```

<xsd:element name="baseElement" type="tBaseElement"/>
<xsd:complexType name="tBaseElement" abstract="true">
  <xsd:sequence>
    <xsd:element ref="documentation" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="extensionElements" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID" use="optional"/>
  <xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>
<xsd:element name="baseElementWithMixedContent"
type="tBaseElementWithMixedContent"/>
  <xsd:complexType name="tBaseElementWithMixedContent" abstract="true"
mixed="true">
    <xsd:sequence>
      <xsd:element ref="documentation" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="extensionElements" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:ID" use="optional"/>
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
  </xsd:complexType>
  <xsd:element name="extensionElements" type="tExtensionElements"/>
  <xsd:complexType name="tExtensionElements">
    <xsd:sequence>
      <xsd:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="documentation" type="tDocumentation"/>
  <xsd:complexType name="tDocumentation" mixed="true">
    <xsd:sequence>
      <xsd:any namespace="##any" processContents="lax" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:ID" use="optional"/>
    <xsd:attribute name="textFormat" type="xsd:string" default="textplain"/>
  </xsd:complexType>

```

Relationship выступает таким элементом, который позволяет выстраивать взаимодействие между артефактами диаграммы. Схема классов элемента Relationship приведена на рис. 4.

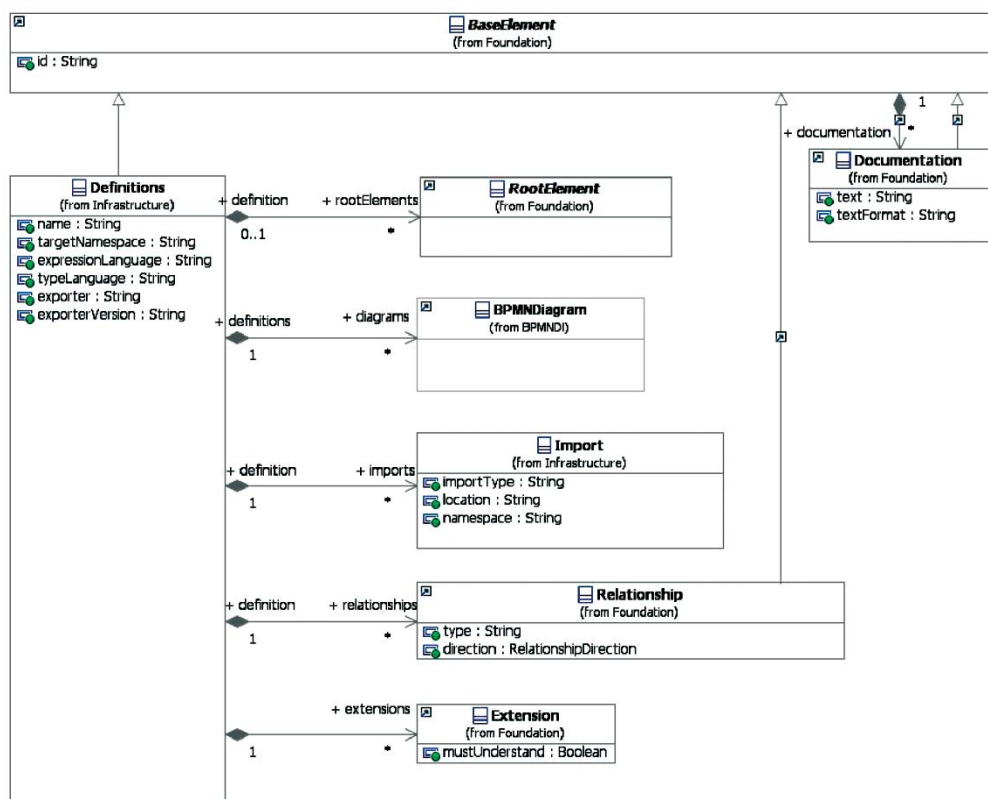


Рис. 4. Схема классов элемента Relationship

Схема XML для элемента Relationship

```

<xsd:element name="relationship" type="tRelationship"/>
<xsd:complexType name="tRelationship">
  <xsd:complexContent>
    <xsd:extension base="tBaseElement">
      <xsd:sequence>
        <xsd:element name="source" type="xsd:QName" minOccurs="1"
maxOccurs="unbounded"/> <xsd:element
  name="target" type="xsd:QName" minOccurs="1"
maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="type" type="xsd:string" use="required"/>
      <xsd:attribute name="direction" type="tRelationshipDirection"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:simpleType name="tRelationshipDirection">

```

```

<xsd:restriction base="xsd:string">
<xsd:enumeration value="None"/>
<xsd:enumeration value="Forward"/>
<xsd:enumeration value="Backward"/>
<xsd:enumeration value="Both"/>
</xsd:restriction>
</xsd:simpleType>
    
```

Для того чтобы описать модель BPMN в формате XML, требуется описать и наполнить данными базовые структуры данных, а затем необходимые стандартные элементы.

Обсуждение полученных результатов. Стандарт BPMN предусматривает возможность моделирования процессов при соответствии требованиям спецификации [30, 31]. Это означает, что все данные о диаграмме программа должна хранить в соответствующем формате XML.

Разрабатывая модели в ПО, полностью соответствующем стандарту BPMN, можно экспортировать созданную модель в формате XML в файл. Если необходимо описать модели вручную или разработать программу для создания и редактирования моделей, то следует руководствоваться официальными документами спецификации BPMN с описанием структур данных и схем XML каждого элемента и пакета. Блок-схема алгоритма реализации описания XML диаграммы BPMN приведена на рис. 5.

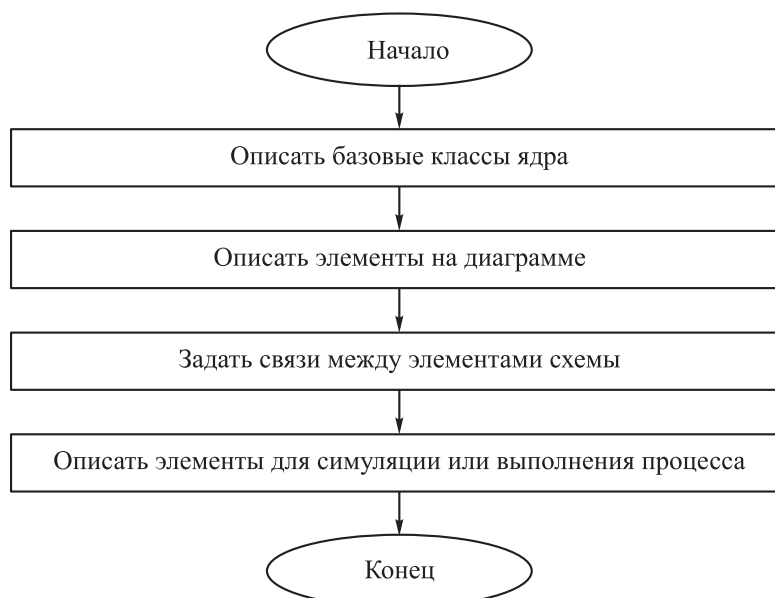


Рис. 5. Обобщенная схема основных этапов алгоритма описания XML диаграммы BPMN

На первом этапе алгоритма необходимо описать базовые классы ядра, затем элементы на диаграмме, такие как задачи, пулы и т. д. Далее необходимо задать связи между элементами схемы соответствующими по смыслу процесса стрелками. После описания базовых элементов и базовых классов ядра необходимо описать события и другие элементы. Указанная методика позволяет шаг за шагом, как конструктор, собирать описание XML исследуемого процесса.

Проведенный анализ дает возможность сформулировать следующую гипотезу визуального системного проектирования: на всех уровнях системного проекта элементы и диаграммы комплексной визуальной модели должны представляться единым унифицированным метаописанием XML, обеспечивающим инкапсуляцию базовых визуальных компонентов в единую замкнутую сеть.

Использование схем XML позволяет разрабатывать собственные программные модели взаимодействия с описанием BPMN (валидаторы, конверторы и визуализаторы процессов предприятия с учетом внутренних правил и регламента), а также собственные симуляторы, реализующие нетипичное поведение в рамках разработанных процессов.

Существенным для реализации указанного подхода является разработка программного средства, реализующего все известные и перспективные методики визуального моделирования на базе единой платформы. В свою очередь, это позволит не только использовать результаты моделирования одного уровня экспертизы при построении моделей другого уровня, но и расширить жизненный цикл модели так, что, проходя каждый новый виток эволюции, развиваться будет созданная в предыдущей итерации модель. Таким образом, будет обеспечено соответствие модели в любой момент времени описываемой системе. Язык I-XML (Intelligence XML), как подмножество XML, для комплексного описания базовых визуальных моделей по уровням экспертизы, на базе которого создан универсальный визуальный редактор VI (Visual Intelligence), предложен в [7, 8].

Заключение. В процессе проведения анализа визуального проектирования сложных систем установлено, что при удобстве и гибкости каждого конкретного метода требуется обращать внимание на возможность взаимодействия этого метода с другими. Единое пространство моделирования позволяет достичь эффективного взаимодействия методов на различных уровнях проектирования. Важно учесть, что модели при взаимодействии должны описываться на метауровне, их реализацию необходимо осуществлять с использованием специальных средств автоматизированной системы проектирования.

С учетом изложенного следует отметить, что создание универсального подхода проектирования, в котором предложены инструменты, необходимые для проектирования на требуемых уровнях детализации, является основной задачей. Следует также разработать автоматизированную среду визуального проектирования, применяющую методику использования специальных средств и гибких инструментов проектирования и предлагающую соответствующие механизмы взаимодействия с концептуально-абстрактными, структурно-функциональными и объектными моделями. Среда позволит осуществить полный цикл проектирования. Таким образом, пользователь получит гибкий инструмент для эффективного решения задач.

Следует отметить, что разработка проприетарного ПО для решения задач взаимодействия с диаграммами XML трудоемка, требует от разработчика опыта решения идентичных задач, что необходимо учитывать при выборе исполнителей подобного технического задания.

ЛИТЕРАТУРА

- [1] Звонкин А.К., Левин Л.А. Сложность конечных объектов и обоснование понятий информации и случайности с помощью теории алгоритмов. *Успехи математических наук*, 1970, т. 25, № 6, с. 85–127.
- [2] Макол Р., ред. Справочник по системотехнике. М., Сов. радио, 1970.
- [3] Бир С. Кибернетика и управление производством. М., Наука, 1965.
- [4] Перегудов Ф.И., Тарасенко Ф.Л. Введение в системный анализ. М., Высш. шк., 1989.
- [5] Квейд Э. Анализ сложных систем. М., Сов. радио, 1969.
- [6] Денисов А.А., Колесников Д.М. Теория больших систем управления. Л., Энергоиздат, 1982.
- [7] Власов А.И. Концепция визуального анализа сложных систем в условиях синхронных технологий проектирования. *Датчики и системы*, 2016, № 8-9, с. 19–25.
- [8] Власов А.И. Применение методов визуального моделирования для формализации конструкторско-технологической информации. *Информатизация образования–2012. Мат. Междунар. науч.-практ. конф.* Орёл, ОГУ, 2012, с. 70–78.
- [9] Кознов Д.В. Языки визуального моделирования: проектирование и визуализация программного обеспечения. СПб., Изд-во СПбГУ, 2004.
- [10] White S.A., Bock C. BPMN 2.0 handbook second edition. Future Strategies Inc., 2011.
- [11] Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. СПб., Питер, 2002.

- [12] Маклаков С.В. Создание информационных систем с AllFusion Modeling Suite. М., Диалог-МИФИ, 2003.
- [13] Иванова Г.С. Технология программирования. М., Изд-во МГТУ им. Н.Э. Баумана, 2002.
- [14] Власов А.И., Журавлева Л.В., Тимофеев Г.Г. Методы генерационного визуального синтеза технических решений в области микронаносистем. *Научное обозрение*, 2013, № 1, с. 107–111.
- [15] Калянов Г.Н. CASE технологии. Консалтинг при автоматизации бизнес-процессов. М., Горячая линия-Телеком, 2000.
- [16] Репин В.В., Елиферов В.Г. Процессный подход к управлению. Моделирование бизнес-процессов. М., Стандарты и качество, 2004.
- [17] Нечаев В.В., Нечаева Н.В. Когнитивно-ментальная информационная деятельность человека. *Информационные и телекоммуникационные технологии*, 2013, № 17, с. 61–74.
- [18] Власов А.И. Системный анализ технологических процессов производства сложных технических систем с использованием визуальных моделей. *Международный научно-исследовательский журнал*, 2013, № 10-2, с. 17–26.
- [19] Власов А.И. Пространственная модель оценки эволюции методов визуального проектирования сложных систем. *Датчики и системы*, 2013, № 9, с. 10–28.
- [20] Шахнов В.А., Власов А.И., Журавлева Л.В. Визуальные методы в условиях синхронных технологий передачи знаний. *Управление качеством инженерного образования. Возможности вузов и потребности промышленности. Тез. докл. междунар. науч.-практ. конф.* М., Изд-во МГТУ им. Н.Э. Баумана, 2016, с. 153–154.
- [21] Vlasov A.I., Demin A.A. Visual methods of formalization of knowledge in the conditions of the synchronous technologies of system engineering. *Proc. CEE-SECR'17*, St. Petersburg, 2017, pp. 1–7. DOI: <https://doi.org/10.1145/3166094.3166098>
- [22] Власов А.И., Журавлева Л.В., Казаков В.В. Применение визуальных инструментов BPMN для моделирования технологической подготовки производства. *Информационные технологии в проектировании и производстве*, 2020, № 1, с. 14–26.
- [23] Grosskopf A., Decker G., Weske M. *The process*. Meghan Kiffer Press, 2009.
- [24] Ryan K.L. Ko, Lee S.S.G., Lee E.W. Business process management (BPM) standards: a survey. *Bus. Process Manag. J.*, 2009, vol. 15, iss. 5. DOI: <https://doi.org/10.1108/14637150910987937>
- [25] Vlasov A.I., Gonoshilov D.S. Simulation of manufacturing systems using BPMN visual tools. *J. Phys.: Conf. Ser.*, 2019, vol. 1353, art. 012043. DOI: <https://doi.org/10.1088/1742-6596/1353/1/012043>
- [26] Conforti R., Dumas M., García-Bañuelos L., et al. BPMN miner: automated discovery of BPMN process models with hierarchical structure. *Inf. Syst. J.*, 2016, vol. 56, pp. 284–303. DOI: <https://doi.org/10.1016/j.is.2015.07.004>

[27] Матвеев А.С., Руденко А.Ю., Прочухан В.В. Разработка рекомендаций перехода от нотации моделирования бизнес-процессов IDEF0 к нотации BPMN. *Бизнес. Образование. Право*, 2016, № 3, с. 176–182.

[28] Потрясаев С.А. Комплексное моделирование сложных процессов на основе нотации BPMN. *Известия высших учебных заведений. Приборостроение*, 2016, т. 59, № 11, с. 913–920.

[29] Федоров И.Г. Моделирование бизнес-процессов в нотации BPMN 2.0. М., МЭСИ, 2013.

[30] Business process model and notation (BPMN).

URL: <https://www.omg.org/spec/BPMN/2.0/PDF> (дата обращения: 28.01.2020).

[31] Власов А.И., Журавлева Л.В., Казаков В.В. Анализ средств разработки визуальных BPMN-моделей сложных систем. *Динамика сложных систем — XXI век*, 2020, т. 14, № 1, с. 5–22. DOI: <https://doi.org/10.18127/j19997493-202001-01>

Власов Андрей Игоревич — канд. техн. наук, доцент кафедры «Проектирование и технология производства электронной аппаратуры» МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5, стр. 1).

Журавлева Людмила Васильевна — канд. техн. наук, доцент кафедры «Проектирование и технология производства электронной аппаратуры» МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5, стр. 1).

Казаков Вадим Вячеславович — студент магистратуры кафедры «Проектирование и технология производства электронной аппаратуры» МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5, стр. 1).

Просьба ссылаться на эту статью следующим образом:

Власов А.И., Журавлева Л.В., Казаков В.В. Методы формализации когнитивной графики и визуальных моделей с использованием схем XML. *Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение*, 2021, № 1 (134), с. 51–77.

DOI: <https://doi.org/10.18698/0236-3933-2021-1-51-77>

**METHODS FOR FORMALIZING COGNITIVE GRAPHICS
AND VISUAL MODELS USING XML SCHEMAS**

A.I. Vlasov

L.V. Zhuravleva

V.V. Kazakov

vlasovai@bmstu.ru

zhuravleva@bmstu.ru

kazakov.vadim@yandex.ru

Bauman Moscow State Technical University, Moscow, Russian Federation

Abstract

The paper analyses methods of formalising cognitive graphics and visual models using promising data storage formats. We describe the primary visual design techniques and note that they appear to be rather disconnected. We show that ensuring the coupling of data and knowledge in visual models featuring various levels of detail is the main problem in integrated usage of visual modelling tools. We analyse approaches to solving the semantic discontinuity problem, that is, provided we meet the condition under which the properties of objects, systems and processes under consideration are only input once, it is necessary to ensure that data from models corresponding to different levels of abstraction (expertise) is interconnected. One should assume that the main drawback of existing approaches to visualising complex systems is that these approaches are fragmented and isolated, which means that they will only be effective locally. The paper proposes several approaches to formalising visual models employing XML schemas, which ensures that development processes concerning visual models of various levels of abstraction are synchronised and interconnected. We use a BPMN (Business, Process, Model and Notation) visual model as an example that shows the principles of representing visual model elements by means of XML schemas. The paper provides a detailed analysis of the principles behind layer interaction in the BPMN model through flexible XML description. We show that the BPMN data structure boasts its own XML schema containing all the parameters of a class or an element. The paper presents several examples and a technique of applying an XML schema to a BPMN model, including a further generalisation aimed at formally representing the process models of complex systems

Keywords

Systems analysis, visual tools, XML schema, Business Process Model and Notation, cognitive graphics

Received 03.03.2020

Accepted 07.12.2020

© Author(s), 2021

Specific results were obtained due to support received from the Ministry of Education and Science of Russian Federation as part of the project titled "Theoretical studies of digital transformation methods concerning core components of micro- and nanosystems" (project no. 0705-2020-0041)

REFERENCES

- [1] Zvonkin A.K., Levin L.A. The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. *Russ. Math. Surv.*, 1970, vol. 25, no. 6, pp. 83–124.
DOI: <http://dx.doi.org/10.1070/RM1970v025n06ABEH001269>

- [2] Machol R.E. System engineering handbook. McGraw Hill, 1965.
- [3] Beer S. Cybernetics and management. English Univ. Press, 1959.
- [4] Peregudov F.I., Tarasenko F.L. Vvedenie v sistemnyy analiz [Introduction to system analysis]. Moscow, Vysshaya shkola Publ., 1989.
- [5] Quade E.S. Analysis for military decisions. Rand McNally, 1964.
- [6] Denisov A.A., Kolesnikov D.M. Teoriya bol'shikh sistem upravleniya [Theory of large control systems]. Leningrad, Energoizdat Publ., 1982.
- [7] Vlasov A.I. The concept of the visual analysis of difficult systems in the conditions of synchronous technologies of designing. *Datchiki i sistemy* [Sensors & Systems], 2016, no. 8-9, pp. 19–25 (in Russ.).
- [8] Vlasov A.I. [Application of methods of visual modeling for formalization of design-technology information]. *Informatizatsiya obrazovaniya–2012. Mat. Mezhdunar. nauch.-prakt. konf.* [Informatization of Education–2012. Proc. Int. Sc.-Pract. Conf.]. Orel, Orel State Univ. Publ., 2012, pp. 70–78 (in Russ.).
- [9] Koznov D.V. Yazyki vizual'nogo modelirovaniya: proektirovanie i vizualizatsiya programmogo obespecheniya [Visual modeling languages: software design and visualization]. St. Petersburg, SPbU Publ., 2004.
- [10] White S.A., Bock C. BPMN 2.0 handbook second edition. Future Strategies Inc., 2011.
- [11] Jacobson I., Booch C., Rumbaugh J. Unified software development process. Addison-Wesley, 1999.
- [12] Maklakov S.V. Sozдание informatsionnykh sistem s AllFusion Modeling Suite [Building information systems with ALLFusion Modeling Suite]. Moscow, Dialog-MIFI Publ., 2003.
- [13] Ivanova G.S. Tekhnologiya programmirovaniya [Programming technology]. Moscow, BMSTU Publ., 2002.
- [14] Vlasov A.I., Zhuravleva L.V., Timofeev G.G. Methods of generation visual synthesis of technical solutions in the sphere of micro-nano-electronic mechanic systems. *Nauchnoe obozrenie*, 2013, no. 1, pp. 107–111 (in Russ.).
- [15] Kalyanov G.N. CASE tekhnologii. Konsalting pri avtomatizatsii biznes-protsessov [CASE technologies. Business process automation consulting]. Moscow, Goryachaya liniya-Telekom Publ., 2000.
- [16] Repin V.V., Eliferov V.G. Protsessnyy podkhod k upravleniyu. Modelirovanie biznes-protsessov [Process approach to management. Modeling of business processes]. Moscow, Standarty i kachestvo Publ., 2004.
- [17] Nechaev V.V., Nechaeva N.V. Cognitive-mental information activity of a person. *Informatsionnye i telekommunikatsionnye tekhnologii* [Information and Telecommunication Technologies], 2013, no. 17, pp. 61–74 (in Russ.).
- [18] Vlasov A.I. System analysis of the technological processes of the complex technical systems with visual models. *Mezhdunarodnyy nauchno-issledovatel'skiy zhurnal* [International Research Journal], 2013, no. 10-2, pp. 17–26 (in Russ.).

- [19] Vlasov A.I. Spatial model assessment of evolution methods of visual design of complex systems. *Datchiki i sistemy* [Sensors & Systems], 2013, no. 9, pp. 10–28 (in Russ.).
- [20] Shakhnov V.A., Vlasov A.I., Zhuravleva L.V. [Visual methods in the conditions of synchronous technologies of transfer of knowledge]. *Upravlenie kachestvom inzhenernogo obrazovaniya. Vozmozhnosti vuzov i potrebnosti promyshlennosti. Tez. dokl. mezhdunar. nauch.-prakt. konf.* [Quality Management in Engineering Education. Capabilities of Universities and Needs of Industry. Abs. Int. Sc.-Pract. Conf.]. Moscow, BMSTU Publ., 2016, pp. 153–154 (in Russ.).
- [21] Vlasov A.I., Demin A.A. Visual methods of formalization of knowledge in the conditions of the synchronous technologies of system engineering. *Proc. CEE-SECR*, St. Petersburg, 2017, art. no. 4, pp. 1–7. DOI: <https://doi.org/10.1145/3166094.3166098>
- [22] Vlasov A.I., Zhuravleva L.V., Kazakov V.V. Application of BPMN visual instruments in modeling technological preparation of production (review). *Informatsionnye tekhnologii v proektirovanii i proizvodstve* [Information Technologies of CAD/CAM/CAE], 2020, no. 1, pp. 14–26 (in Russ.).
- [23] Grosskopf A., Decker G., Weske M. *The process*. Meghan Kiffer Press, 2009.
- [24] Ryan K.L. Ko, Lee S.S.G., Lee E.W. Business process management (BPM) standards: a survey. *Bus. Process Manag. J.*, 2009, vol. 15, iss. 5. DOI: <https://doi.org/10.1108/14637150910987937>
- [25] Vlasov A.I., Gonoshilov D.S. Simulation of manufacturing systems using BPMN visual tools. *J. Phys.: Conf. Ser.*, 2019, vol. 1353, art. 012043. DOI: <https://doi.org/10.1088/1742-6596/1353/1/012043>
- [26] Conforti R., Dumas M., García-Bañuelos L., et al. BPMN miner: automated discovery of BPMN process models with hierarchical structure. *Inf. Syst. J.*, 2016, vol. 56, pp. 284–303. DOI: <https://doi.org/10.1016/j.is.2015.07.004>
- [27] Matveev A.S., Rudenko A.Yu., Prochukhan V.V. Development of recommendations for transition from IDEF0 notation of business processes modeling to BPMN notation. *Biznes. Obrazovanie. Pravo*, 2016, no. 3, pp. 176–182 (in Russ.).
- [28] Potryasaev S.A. Integrated modeling of complex processes based on BPMN notation. *Izvestiya vysshikh uchebnykh zavedeniy. Priborostroenie* [Journal of Instrument Engineering], 2016, vol. 59, no. 11, pp. 913–920 (in Russ.).
- [29] Fedorov I.G. *Modelirovanie biznes-protsessov v notatsii BPMN 2.0* [Modelling business processes in the BPMN 2.0 notation]. Moscow, MESI Publ., 2013.
- [30] Business process model and notation (BPMN). Available at: <https://www.omg.org/spec/BPMN/2.0/PDF> (accessed: 28.01.2020).
- [31] Vlasov A.I., Zhuravleva L.V., Kazakov V.V. Application of BPMN visual instruments in modeling technological preparation of production. *Dinamika slozhnykh sistem — XXI vek* [Dynamics of Complex Systems — XXI Century], 2020, vol. 14, no. 1, pp. 5–22 (in Russ.). DOI: <https://doi.org/10.18127/j19997493-202001-01>

Vlasov A.I. — Cand. Sc. (Eng.), Assoc. Professor, Department of Electronic Equipment Design and Technology, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5, str. 1, Moscow, 105005 Russian Federation).

Zhuravleva L.V. — Cand. Sc. (Eng.), Assoc. Professor, Department of Electronic Equipment Design and Technology, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5, str. 1, Moscow, 105005 Russian Federation).

Kazakov V.V. — Master's Degree Student, Department of Electronic Equipment Design and Technology, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5, str. 1, Moscow, 105005 Russian Federation).

Please cite this article in English as:

Vlasov A.I., Zhuravleva L.V., Kazakov V.V. Methods for formalizing cognitive graphics and visual models using XML schemas. *Herald of the Bauman Moscow State Technical University, Series Instrument Engineering*, 2021, no. 1 (134), pp. 51–77 (in Russ.). DOI: <https://doi.org/10.18698/0236-3933-2021-1-51-77>