



Международный журнал информационных технологий и энергоэффективности

Сайт журнала: <http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.89

## ФОРМАТ БАЗЫ ЗНАНИЙ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ДЛЯ ИНТЕЛЛЕКТУАЛЬНОГО УПРАВЛЕНИЯ КОМПЛЕКСНЫМИ РИСКАМИ В СЛОЖНЫХ ОРГАНИЗАЦИОННО-ТЕХНИЧЕСКИХ СИСТЕМАХ<sup>1</sup>

Сеньков А.В.

Филиал ФГБОУ ВО "НИУ "МЭИ" в г. Смоленске, Россия (214013, г. Смоленск, Энергетический проезд, дом 1); e-mail: a.v.senkov@mail.ru

Статья посвящена разработке формата базы знаний системы поддержки принятия решений для интеллектуального управления комплексными рисками сложных организационно-технических системах. Предлагаемая структура обладает достаточной гибкостью, представлены варианты структуры для реализации в рамках базы данных, а также для реализации в виде открытой xsd-схемы, обеспечивающей возможность свободного обмена знаниями по интеллектуальному управлению рисками между всеми заинтересованными сторонами.

Ключевые слова: интеллектуальное управление рисками, база знаний, xsd.

## KNOWLEDGE BASE FORMAT OF THE DECISION-MAKING SUPPORT SYSTEM FOR INTELLIGENT MANAGEMENT OF INTEGRATED RISKS IN COMPLEX ORGANIZATIONAL AND TECHNICAL SYSTEMS

Senkov A.V.

Smolensk Branch of the National Research University "Moscow Power Engineering Institute", Russia (214013, Smolensk, street Ehnergeticheskij, 1); e-mail: a.v.senkov@mail.ru

The article is devoted to the development of the format of the knowledge base of the decision-making support system for the intelligent management of integrated risks of complex organizational and technical systems. The proposed structure has sufficient flexibility, provides options for the structure for implementation within the database, and for implementation in the form of an open xsd-scheme that provides the opportunity for free knowledge exchange on intelligent risk management among all stakeholders.

Key words: intelligent risk management, knowledge base, xsd.

В настоящее время ведутся активные работы по разработки методов, моделей и способов управления рисками в сложных организационно-технических системах [1, 2, 3]. Особенностью таких методов, моделей и способов является их ограниченность,

<sup>1</sup> Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 16-37-60059 мол\_а\_дк

нацеленность на решение строго определенной задачи управления рисками. Анализ баз цитирований Scopus и РИНЦ позволяет говорить о том, что до 80% интеллектуальных моделей, предлагаемых в таких материалах имеют своей целью оценивание рисков, при этом оценивание – далеко не самый значимый этап управления рисками.

В то же время в работе [4] предлагается комплексный подход к управлению рисками, основанный на рассмотрении как прелиминарного (заблаговременного) управления рисками, так и прецедентного управления рисками (управление последствиями). Кроме того, подход предполагает рассмотрение трёх аспектов управления рисками: процессного, структурного и системного. Указанные три аспекта позволяют наиболее полно идентифицировать, проанализировать, провести оценивание рисков и выработать решения по управлению ими.

В работе [5] предложена графическая нотация для представления процесса управления комплексными рисками. Такая нотация обеспечивает наглядное представление знаний о рисках, полученных в рамках разрабатываемого подхода. При этом следует учитывать, что знания как в форме фактов, так и в форме правил являются краеугольным камнем разработки систем поддержки принятия решений. Задачам разработки таких систем, функционирующих в различных условиях посвящено множество работ [6, 7], однако, в таких работах, как правило, мало внимания уделяется переносу знаний.

В качестве форматов переноса знаний в последнее время часто используются открытые форматы типа XML [8, 9]. Для разработки формата XML требуется разработать xsd-схему, основанную на понимании структуры передаваемых данных.

#### **Структура знаний в сфере управления рисками**

В рамках [4] выделяются следующие сущности в сфере управления рисками, знания о которых должны храниться, обрабатываться и передаваться.

Риск-ситуация –  $RS_i$  –  $i$ -я риск-ситуация  $i \in 1, \dots, i'$ ,  $i'$  – количество риск-ситуаций.

Риск-событие –  $RE_j$  –  $j$ -е риск-событие,  $j \in 1, \dots, j'$ ,  $j'$  – количество риск-событий.

Источник риска –  $SR_k$  –  $m$ -й источник риска,  $k \in 1, \dots, k'$ ,  $k'$  – количество источников рисков.

Событие –  $E_l$  –  $l$ -й источник риска,  $l \in 1, \dots, l'$ ,  $l'$  – количество событий.

Элемент системы –  $SE_m$  –  $m$ -й источник риска,  $m \in 1, \dots, m'$ ,  $m'$  – количество элементов системы.

Элемент внешней среды –  $EE_n$  –  $n$ -й источник риска,  $n \in 1, \dots, n'$ ,  $n'$  – количество элементов внешней среды.

Прелиминарное мероприятие –  $BMes_o$  –  $o$ -е предиктивное мероприятие,  $o \in 1, \dots, o'$ ,  $o'$  – количество предиктивных мероприятий.

Прецедентное мероприятие –  $AMes_p$  –  $p$ -е прецедентное мероприятие,  $p \in 1, \dots, p'$ ,  $p'$  – количество предиктивных мероприятий.

Знания об этих сущностях в соответствии с [10] представляют собой базу фактов. Базу же правил представляют непосредственно диаграммы, описанные в [5]:

1. диаграмма процессного аспекта рисков;
2. диаграмма структурного аспекта рисков;

3. диаграмма системного аспекта рисков;
4. диаграмма риск-ситуаций;
5. диаграмма прецедентного управления рисками;
6. диаграмм предварительного управления рисками.

Для хранения указанных фактов и правил в базе знаний предлагается использовать следующую её структуру (рисунок 1).

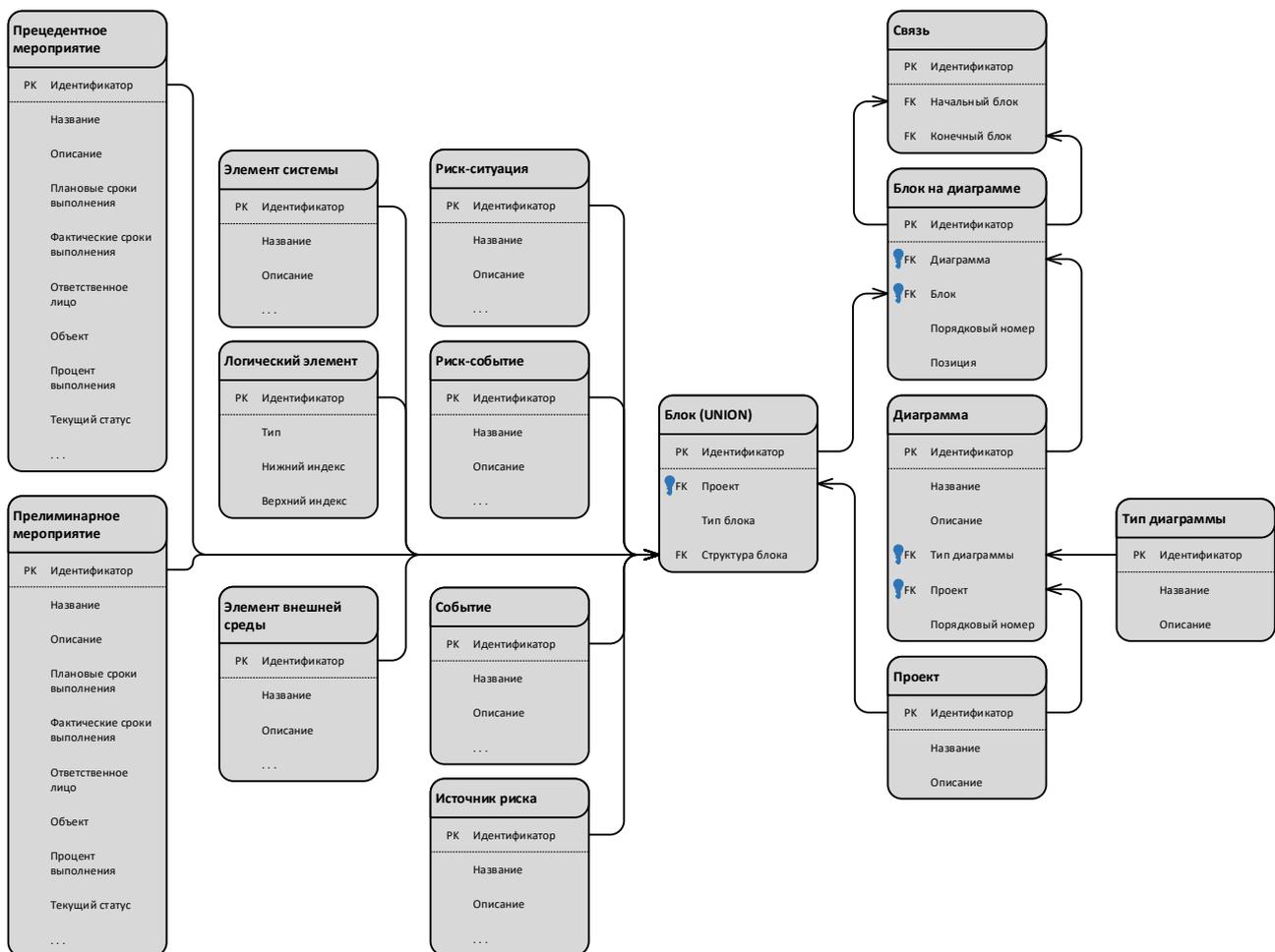


Рисунок 1 – Структура базы знаний для управления рисками

Представленная структура обладает следующими достоинствами:

1. она адаптирована для хранения в широко распространенных СУБД, поскольку представлена в формате ER-диаграммы;
2. полностью отражает все элементы нотации рисков, а значит содержит все необходимые для накопления, хранения и передачи факты и правила для управления рисками;
3. может быть легко конвертирована в открытый формат обмена данными, что обеспечит широкие возможности по обмену знаниями между всеми заинтересованными сторонами.

**XSD-схема для обмена знаниями между системами поддержки принятия решений**

XSD-схема для обмена знаниями основывается на представленной на рисунке 1 структуре базы знаний и включает следующие типы элементов:

1. базовые типы – типы элементов, описывающие базовые типы данных и знаний, задающие форматы таких описаний;
2. справочные типы – типы элементов, позволяющие задать справочники и классификаторы, единые для всех пользователей открытого формата;
3. типы фактов – типы элементов, задающих факты базы знаний;
4. типы правил – типы элементов, задающих правила работы над фактами.

Рассмотрим перечисленные типы подробнее.

### **Базовые типы**

К базовым типам относим:

- уникальный идентификатор (UUID);
- название (Name);
- описание (Description);
- положение (Position);
- точка (Point);
- массив точек (PointArray);
- срок (Period);
- процент (Percent);
- статус (Status).

**Уникальный идентификатор** используется для уникальной идентификации всех фактов и правил в рамках базы знаний, а также между базами знаний. Описывается в виде.

```
<xs:simpleType name="uuid">
  <xs:restriction base="xs:string">
    <xs:pattern value="[0-9a-fA-F]{32}"/>
  </xs:restriction>
</xs:simpleType>
```

**Название** используется для задания названий элементам, представляет собой строку из 50 символов. Описывается в виде.

```
<xs:simpleType name="Name">
  <xs:restriction base="xs:string">
    <xs:minLength value="0"/>
    <xs:maxLength value="50"/>
  </xs:restriction>
</xs:simpleType>
```

**Описание** используется для описания или комментирования объектов, представляет собой строку. Описывается в виде.

```
<xs:simpleType name="Description">
  <xs:restriction base="xs:string">
    <xs:minLength value="0"/>
  </xs:restriction>
</xs:simpleType>
```

**Положение** предназначено для задания координат блока на диаграмме. Представляет собой пару параметров «x» и «y», каждый из которых целое число, описывается в следующем виде.

```
<xs:complexType name="Position">
  <xs:sequence>
    <xs:element name="x" type="xs:integer"/>
    <xs:element name="y" type="xs:integer"/>
  </xs:sequence>
</xs:complexType>
```

**Точка** используется для задания параметра типа точка. Представляет собой пару параметров «x» и «y», каждый из которых целое число, описывается в следующем виде.

```
<xs:complexType name="Point">
  <xs:sequence>
    <xs:element name="x" type="xs:integer"/>
    <xs:element name="y" type="xs:integer"/>
  </xs:sequence>
</xs:complexType>
```

**Массив точек** используется для задания нечеткого значения. Представляет собой неограниченный массив точек, описывается в следующем виде.

```
<xs:complexType name="PointArray">
  <xs:sequence>
    <xs:element name="p" type="Point" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

**Период** используется для задания сроков выполнения мероприятия. Представляет собой комбинацию из даты/времени начала и даты/времени окончания процесса, описывается в следующем виде.

```
<xs:complexType name="Period">
  <xs:sequence>
    <xs:element name="begin" type="xs:dateTime"/>
    <xs:element name="end" type="xs:dateTime"/>
  </xs:sequence>
</xs:complexType>
```

**Процент** используется для задания процента выполнения мероприятия. Представляет собой целое число от 0 до 100.

```
<xs:simpleType name="Percent">
  <xs:restriction base="xs:integer">
    <xs:minExclusive value="0"/>
    <xs:maxExclusive value="100"/>
  </xs:restriction>
</xs:simpleType>
```

**Статус** используется для задания статуса выполнения мероприятия. Представляет собой список выбора: «new» (Новая), «inProcess» (В процессе), «complete» (За-вершена), «reject» (Отклонена), описывается следующим образом.

```
<xs:simpleType name="Status">
  <xs:restriction base="xs:string">
    <xs:enumeration value="new"/>
    <xs:enumeration value="inProcess"/>
    <xs:enumeration value="complete"/>
    <xs:enumeration value="reject"/>
  </xs:restriction>
</xs:simpleType>
```

### Справочные типы

К справочным типам относим:

- тип диаграммы (DiagramType);
- тип блока (BlockType);
- тип логического элемента (LogicElementType).

**Тип диаграммы** используется для задания справочника типов диаграмм. Представляет собой структуру из полей: «id» (Идентификатор), «name» (Название), «description» (Описание). Поле «description» не является обязательным. Задается следующим образом.

```
<xs:element name="DiagramType">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="id" type="uuid"/>
      <xs:element name="name" type="Name"/>
      <xs:element name="description" type="Description" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**Тип блока** используется для задания типа блока на диаграмме. Представляет собой список выбора: «RiskSituation» (Риск-ситуация), «RiskEvent» (Риск-событие), «Event» (Событие), «RiskSource» (Источник риска), «CaseAction» (Прецедентное мероприятие), «PremilinaryAction» (Прелиминарное мероприятие), «InternalElement» (Элемент системы), «ExternalElement» (Элемент внешней среды), «LogicElement» (Логический элемент). Задается следующим образом.

```
<xs:simpleType name="BlockType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="RiskSituation"/>
    <xs:enumeration value="RiskEvent"/>
    <xs:enumeration value="Event"/>
    <xs:enumeration value="RiskSource"/>
    <xs:enumeration value="CaseAction"/>
    <xs:enumeration value="PremilinaryAction"/>
    <xs:enumeration value="InternalElement"/>
    <xs:enumeration value="ExternalElement"/>
    <xs:enumeration value="LogicElement"/>
  </xs:restriction>
```

```
</xs:simpleType>
```

Тип логического элемента используется для задания типа логического блока. Представляет собой список выбора. Описывается следующим образом.

```
<xs:simpleType name="LogicElementType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="And"/>
    <xs:enumeration value="Or"/>
    <xs:enumeration value="Not"/>
    <xs:enumeration value="Xor"/>
    <xs:enumeration value="Next"/>
    <xs:enumeration value="Last"/>
    <xs:enumeration value="AlwaysInFuture"/>
    <xs:enumeration value="AlwaysInPast"/>
    <xs:enumeration value="SometimeInFuture"/>
    <xs:enumeration value="SometimeInPast"/>
    <xs:enumeration value="Until"/>
    <xs:enumeration value="Unless"/>
    <xs:enumeration value="Since"/>
    <xs:enumeration value="Zince"/>
    <xs:enumeration value="Before"/>
    <xs:enumeration value="After"/>
    <xs:enumeration value="Meets"/>
    <xs:enumeration value="MetBy"/>
    <xs:enumeration value="Overlaps"/>
    <xs:enumeration value="OverlappedBy"/>
    <xs:enumeration value="During"/>
    <xs:enumeration value="Includes"/>
    <xs:enumeration value="Starts"/>
    <xs:enumeration value="StartedBy"/>
    <xs:enumeration value="Finishes"/>
    <xs:enumeration value="FinishedBy"/>
    <xs:enumeration value="Equals"/>
  </xs:restriction>
</xs:simpleType>
```

#### Типы фактов

К типам фактов относим:

- риск-ситуацию (RiskSituation);
- риск-событие (RiskEvent);
- событие (Event);
- источник риска (RiskSource);
- прецедентное мероприятие (CaseAction);
- прелиминарное мероприятие (PreliminaryAction);
- элемент системы (InternalElement);
- элемент внешней среды (ExternalElement);
- логический элемент (LogicElement).

Все перечисленные блоки (кроме блока «Логический элемент») сами по себе являются фактами и используются в дальнейшем при построении правил. Описываются следующим образом.

```
<xs:element name="RiskSituation">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="id" type="uuid"/>
      <xs:element name="name" type="Name"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
<xs:element name="description" type="Description" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="RiskEvent">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="id" type="uuid"/>
      <xs:element name="name" type="Name"/>
      <xs:element name="description" type="Description" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="Event">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="id" type="uuid"/>
      <xs:element name="name" type="Name"/>
      <xs:element name="description" type="Description" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="RiskSource">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="id" type="uuid"/>
      <xs:element name="name" type="Name"/>
      <xs:element name="description" type="Description" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="CaseAction">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="id" type="uuid"/>
      <xs:element name="name" type="Name"/>
      <xs:element name="description" type="Description" minOccurs="0"/>
      <xs:element name="plan" type="Period"/>
      <xs:element name="actualPeriod" type="Period"/>
      <xs:element name="executor" type="Name"/>
      <xs:element name="object" type="Name"/>
      <xs:element name="percent" type="Percent"/>
      <xs:element name="status" type="Status"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="PremilinaryAction">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="id" type="uuid"/>
      <xs:element name="name" type="Name"/>
      <xs:element name="description" type="Description" minOccurs="0"/>
      <xs:element name="plan" type="Period"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
<xs:element name="actualPeriod" type="Period"/>
<xs:element name="executor" type="Name"/>
<xs:element name="object" type="Name"/>
<xs:element name="percent" type="Percent"/>
<xs:element name="status" type="Status"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="InternalElement">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="id" type="uuid"/>
      <xs:element name="name" type="Name"/>
      <xs:element name="description" type="Description" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="ExternalElement">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="id" type="uuid"/>
      <xs:element name="name" type="Name"/>
      <xs:element name="description" type="Description" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="LogicElement">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="id" type="uuid"/>
      <xs:element name="type" type="LogicElementType"/>
      <xs:element name="minValue" type="xs:integer" minOccurs="0"/>
      <xs:element name="maxValue" type="xs:integer" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

### Типы правил

К типам правил относим:

- проект (Project);
- диаграмма (Diagram);
- соединение (Connection);
- блок в диаграмме (DiagramBlock);
- факт (Block).

**Проект** используется для описания проекта. Содержит в себе основные параметры проекта: «id» (Идентификатор), «name» (Название), «description» (Описание). А также следующие списки: «DiagramType» (Справочник типов диаграмм), «Block» (Список всех блоков проекта), «Diagram» (Список всех диаграмм проекта). Описывается следующим образом.

```
<xs:element name="Project">
```

```
<xs:complexType>
  <xs:sequence>
    <xs:element name="id" type="uuid"/>
    <xs:element name="name" type="Name"/>
    <xs:element name="description" type="Description" minOccurs="0"/>
    <xs:element ref="DiagramType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="Block" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="Diagram" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
```

**Диаграмма** используется для описания диаграммы в проекте. Содержит в себе основные параметры диаграммы: «id» (идентификатор), «name» (название), «description» (описание), «number» (порядковый номер), «diagramTypeId» (идентификатор типа диаграммы из справочника). А также следующие списки: «DiagramBlock» (список блоков), «Connection» (список связей). Описывается следующим образом.

```
<xs:element name="Diagram">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="id" type="uuid"/>
      <xs:element name="name" type="Name"/>
      <xs:element name="description" type="Description" minOccurs="0"/>
      <xs:element name="number" type="xs:integer"/>
      <xs:element name="diagramTypeId" type="uuid"/>
      <xs:element ref="DiagramBlock" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="Connection" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**Связь** используется для описания связи на диаграмме. Представляет собой структуру из параметров объекта: «id» (Идентификатор), «beginDiagramBlockId» (Идентификатор блока на диаграмме, из которого выходит связь), «endDiagramBlockId» (Идентификатор блока на диаграмме, в который входит связь). Описывается следующим образом.

```
<xs:element name="Connection">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="id" type="uuid"/>
      <xs:element name="beginDiagramBlockId" type="uuid"/>
      <xs:element name="endDiagramBlockId" type="uuid"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**Блок в диаграмме** используется для описания блока на диаграмме. Представляет собой структуру из параметров объекта: «id» (Идентификатор), «blockId» (Идентификатор блока из списка блоков проекта), «number» (Порядковый номер блока в диаграмме), «position» (Расположение элемента на диаграмме).

```
<xs:element name="DiagramBlock">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="id" type="uuid"/>
      <xs:element name="blockId" type="uuid"/>
      <xs:element name="number" type="xs:integer"/>
      <xs:element name="position" type="Position"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

**Факт** используется для описания факта в проекте. Содержит в основные параметры проекта: «id» (Идентификатор), «type» (Тип объекта). Описывается следующим образом.

```
<xs:element name="Block">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="id" type="uuid"/>
      <xs:element name="type" type="BlockType"/>
      <xs:element ref="RiskSituation" minOccurs="0"/>
      <xs:element ref="RiskEvent" minOccurs="0"/>
      <xs:element ref="Event" minOccurs="0"/>
      <xs:element ref="RiskSource" minOccurs="0"/>
      <xs:element ref="CaseAction" minOccurs="0"/>
      <xs:element ref="PremilinaryAction" minOccurs="0"/>
      <xs:element ref="InternalElement" minOccurs="0"/>
      <xs:element ref="ExternalElement" minOccurs="0"/>
      <xs:element ref="LogicElement" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Предложенные в настоящей статье структура базы знаний и XSD-схема для передачи знаний в открытом формате XML обладают следующими достоинствами. Структура базы знаний:

1. адаптирована для хранения в широко распространенных СУБД, поскольку представлена в формате ER-диаграммы;
2. полностью отражает все элементы нотации рисков, а значит содержит все необходимые для накопления, хранения и передачи факты и правила для управления рисками;
3. может быть легко конвертирована в открытый формат обмена данными, что обеспечит широкие возможности по обмену знаниями между всеми заинтересованными сторонами.

В свою очередь, XSD-схема

1. позволяет выполнять обмен знаниями между различными базами знаний;
2. обеспечивает валидацию полученных знаний;
3. проста для использования с технологической точки зрения.

### Список литературы

1. Борисов В.В., Сеньков А.В., Интеллектуальное управление рисками в топливно-энергетическом комплексе. – Смоленск: Универсум, 2015

2. Канев В.С., Шевцова Ю.В. Основы моделирования и управления операционными рисками в электронной коммерции и телекоммуникациях. – М.: Горячая линия – Телеком, 2015
3. Астахов А.М. Искусство управления информационными рисками.– М.: ДМК Пресс 2010
4. Сеньков А.В. Управление рисками: интеллектуальные модели, методы, средства. – Смоленск: Универсум, 2016
5. Сеньков А.В., Графическая нотация для представления процесса управления комплексными рисками // Современные наукоемкие технологии. – 2016. – № 12-1 . – С. 72-81
6. Vagin V.N., Ereemeev A.P. Some Basic Principles of Design of Intelligent Systems for Supporting Real-Time Decision Making // J. of Computer and System Sciences International, Vol. 40, No. 6, 2001, pp. 953-961.
7. Ereemeev A.P., Vagin V.N. A Real Time Decision Support System for Monitoring and Management of a Complex Object Using Parallel Processing // Proc. of the IEEE Int. Conf. Artificial Intelligent Systems IEEE AIS'02, Sept. 5-10, 2002, Divnomorskoe, Russia, pp. 139-144
8. W.M.P. van der Aalst, H.M.W. Verbeek, and A. Kumar. Verification of XRL: An XML-based Workflow Language. In W. Shen, Z. Lin, J.P. Barthes, and M. Kamel, editors, Proceedings of the 6th International Conference on CSCW in Design, pages 427-432. NRC Research Press, Ottawa, Canada, 2001
9. W.M.P. van der Aalst and A. Kumar. XML Based Schema Definition for Support of Inter-organizational Workflow. Information Systems Research, 14(1):23-46, 2003
10. Пospelov Д.А. Искусственный интеллект. Справочник. Книга 2. Модели и методы. М.: Радио и связь, 1990.

#### References

1. Borisov VV, Senkov AV, Intellectual Risk Management in the Fuel and Energy Complex. - Smolensk: Universum, 2015
  2. Kanev VS, Shevtsova Yu.V. Basics of modeling and management of operational risks in e-commerce and telecommunications. - М.: Hot line - Telecom, 2015
  3. Astakhov A.M. The art of managing information risks .- Moscow: DМК Press 2010
  4. Senkov A.V. Risk management: intellectual models, methods, tools. - Smolensk: Universum, 2016
  5. Senkov AV, Graphical notation for the presentation of the process of managing complex risks // Modern high technology. - 2016. - No. 12-1. - P. 72-81
  6. Vagin V.N., Ereemeev A.P. Some Basic Principles of Design of Intelligent Systems for Supporting Real-Time Decision Making // J. of Computer and System Sciences International, Vol. 40, No. 6, 2001, pp. 953-961.
  7. Ereemeev A.P., Vagin V.N. A Real Time Decision Support System for Monitoring and Management of a Complex Object Using Parallel Processing // Proc. of the IEEE Int. Conf. Artificial Intelligent Systems IEEE AIS'02, Sept. 5-10, 2002, Divnomorskoe, Russia, pp. 139-144
  8. W.M.P. van der Aalst, H.M.W. Verbeek, and A. Kumar. Verification of XRL: An XML-based Workflow Language. In W. Shen, Z. Lin, J.P. Barthes, and M. Kamel, editors, Proceedings of the 6th International Conference on CSCW in Design, pages 427-432. NRC Research Press, Ottawa, Canada, 2001
  9. W.M.P. van der Aalst and A. Kumar. XML Based Schema Definition for Support of Inter-organizational Workflow. Information Systems Research, 14 (1): 23-46, 2003
  10. Pospelov DA Artificial Intelligence. Directory. Book 2. Models and methods. М.: Radio and Communication, 1990.
-