УДК 621.395

# Базовая онтология неспецифических сущностей BONE и её использование для построения информационных систем

А.Г. Марчук, П.А. Марчук

В работе рассмотрены принципы создания онтологии неспецифических сущностей Basic Ontology for Non-specific Entities (BONE), некоторые особенности её устройства и использование этой онтологии при создании баз данных и информационных систем. Отличительной особенностью представленной онтологии является систематическое использование так называемых составных отношений. Это позволяет атрибутировать не только описываемые объекты, но и отношения между ними. Онтология BONE предназначена для структурирования информации в рамках подходов и технологий Semantic Web.

*Ключевые слова*: BONE, онтология, Semantic Web, информационные системы.

## 1. Введение

В Институте систем информатики СО РАН более 10 лет ведутся фундаментальные и прикладные работы по методам построения и построению архивных систем [1–3]. Был выработан подход, отличающийся тем, что кроме хранения электронных образов документов создаётся база данных, включающая в себя и данные о документах и данные о различных ассоциированных с документами сущностями, такими как персоны, организации, географические объекты. Документы «привязываются» к базе данных через определённые отношения, что позволяет иметь удобный доступ к документам архива как в режиме поиска, так и при анализе изучаемой темы. Такой подход, основанный на фиксации фактов, был назван фактографическим.

Другой особенностью подхода является использование концепции и стандартов Semantic Web [4]. Это дало гибкую систему структуризации в виде формата RDF [5] и средства формального описания данных – языка OWL [6]. Кроме того, стало возможным создавать распределённые и двухуровневые информационные системы. Распределённость относится к базам данных, реализованным в виде RDF-документов и базам документов, формируемых в виде кассет [7]. Пользователи получают возможность выстраивать и контролировать свою базу данных и документов, при этом «встраивать» её в общее информационное пространство. Двухуровневое построение информационных систем означает, что распределённая база данных и документов может использоваться как поле фактов для тематических сайтов и других ресурсов и этот информационный слой не требует дублирования. Такой подход даёт решение для известной проблемы, которая заключается в следующем. В большинстве развитых информационных систем есть данные о персонах, их коммуникационных координатах, об организациях и т.д. При этом каждая из таких ИС имеет свою, не пересекающуюся с другими базу данных, что приводит к необходимости вносить эти данные через регистрацию и другие формы заполнения, а изменения в данных и расширения базы данных существенно затруднены.

Ключевую роль в предложенном подходе играет формальная спецификация данных – онтология. Поскольку речь идёт о данных «обыденной» природы, лишенных предметной специфики, онтология была названа Basic Ontology for Non-specific Entities (BONE). В некоторой части BONE имеет соответствия в известных онтологиях и словарях (vocabularies), однако в целостном виде подобное онтологическое построение авторам не известно. Онтология требуется для задания системы структуризации данных и используется практически во всех компонентах созданного программного обеспечения. Использование онтологии позволяет сделать программы более универсальными и более компактными.

## 1. Основные принципы формирования онтологии

Онтология фиксирует набор сущностных классов и набор отношений между ними. Здесь мы следуем традиционному подходу к классификациям, известному как Entities – Relationships (ER). Кроме того, сущностные классы, а иногда и отношения выстраиваются в иерархию по соответствию класс – подкласс и отношение – подотношение с наследованием свойств. Наиболее часто используемым формализмом для определения онтологий является язык Web Ontology Language (OWL) [6].

Главным критерием, который можно предъявить к системе структуризации, является адекватность системы объекту описания. При этом адекватность, как правило, вступает в противоречие с общностью описания. Например, территориальное разбиение страны на административные единицы может быть самым разным: области, штаты, графства, кантоны и др., но для общности желательно иметь единое понятие, соответствующее такому делению, не зависящее от государственных особенностей. Первый использованный принцип – онтология должна вводить минимальное число понятий, достаточных для описания объектов мира с требуемой для работы детальностью. Система структуризации не должна иметь прямой зависимости от времени рассмотрения данных, географического положения, культурных и национальных особенностей.

Известные онтологии не всегда соответствуют этому критерию, причём в самых простых вопросах. В качестве примера можно привести довольно часто встречаемые поля *возраст*, или *число жителей*, которые явно зависят от времени записи информации.

Для формулирования других принципов нам потребуется использовать особенности использованной системы структуризации – формата RDF [5]. RDF описывает ориентированный граф, состоящий из двух видов вершин – сущностей и значений (или атрибутов). Описание является неупорядоченным набором высказываний, каждое из которых представляет триплет: субъект – предикат – объект. Причём в графе предикату сопоставляется идентифицированная дуга, субъект соответствует идентифицированной сущности, объект соответствует либо идентифицированной сущности, либо значению. Графически, эти два вида высказываний выглядят следующим образом:

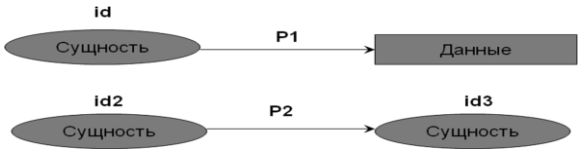


Рис 1. Два вида высказываний, используемых
в описании системы структуризации – формата RDF

Общий же граф формируется из отдельных высказываний путём «склейки» одинаково идентифицированных сущностей. На рис. 2 изображён фрагмент данных, описывающих персону, организацию и отношение между ними.

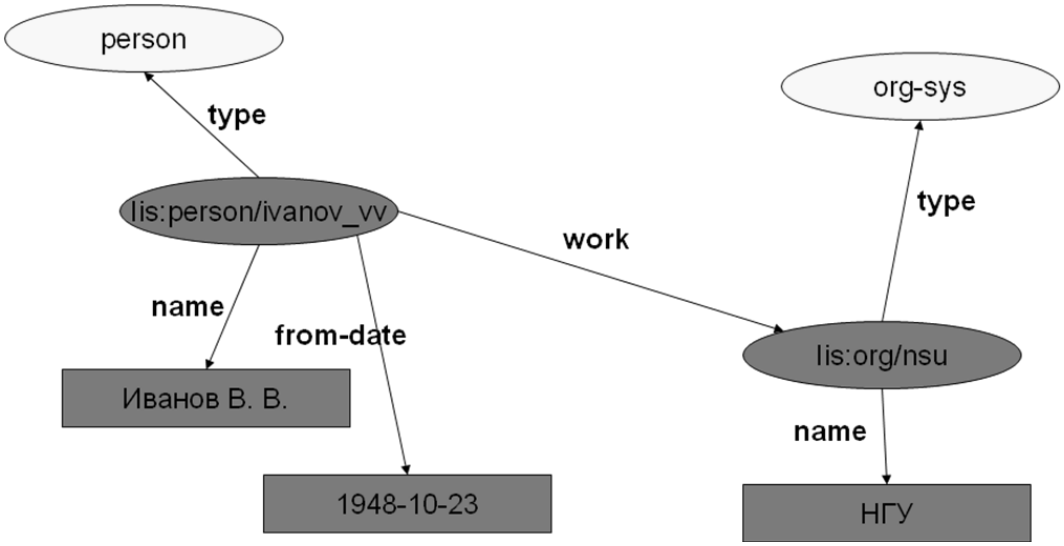


Рис. 2. Фрагмент данных, описывающих персону, организацию и отношение между ними

Легко заметить, что для приведённого примера описание можно свести к записям реляционных таблиц. Записью назовём фрагмент графа, состоящий из одной вершины сущности и множества исходящих дуг. Причём дуги, ведущие к значениям, также включаются в запись, а дуги, ведущие к другим сущностным вершинам, остаются ссылками на эти вершины. Последнее соответствует понятию external key в реляционных базах данных. Для установления соответствия табличным построениям потребуем то, чтобы из узла не выходило больше одной дуги с заданным идентификатором. Кроме того, потребуем, чтобы идентифицированная дуга в онтологии была бы либо предикатом, ведущим к значению (`DatatypeProperty`), либо дугой, ссылающейся на другую сущность (`ObjectProperty`). Эти два требования позволяют реализовывать RDF-базу данных средствами реляционных таблиц.

Есть, правда, особенность, препятствующая такой реализации. Это языковая множественность полей данных. В стандарте RDF поля данных могут иметь языковый спецификатор `xml:lang`, определяющий язык для интерпретации значения. Например, во фрагменте RDF-кода заданы имена персоны на разных языках:

```
<fog:Person rdf:about=”p2987349”>
   <fog:name xml:lang=”ru”>Иванов Иван Иванович</fog:name>
   <fog:name xml:lang=”en”>Ivanov, Ivan Ivanovich</fog:name>
</fog:Person>
```

Такую возможность мы в своей системе структуризации оставляем, поскольку это соответствует задаче адекватного отражения объектов реального мира.

Также можно обратить внимание в примере на прямую связь между персоной и организацией – дугу, идентифицированную как work. Для множества онтологических построений это нормальная конструкция. Однако с ней связаны существенные проблемы. Во-первых, подобных связей (типа «работа») может быть несколько и это противоречит единственности одноимённых полей для записей. Во-вторых, отношения, как правило, имеют атрибуты, существенные для описания. Например, отношение «работа» может иметь такие атрибуты, как «дата начала», «дата конца», «должность» и др. Выходом из обоих затруднений является реализация отношений в виде псевдосущностей, то есть, используя в качестве отношения узлы, из которых прямые ссылки ведут к объектам, между которыми устанавливается отношение. Рис. 3 является модификацией рис. 2, но со сформированной по такому положению связью.

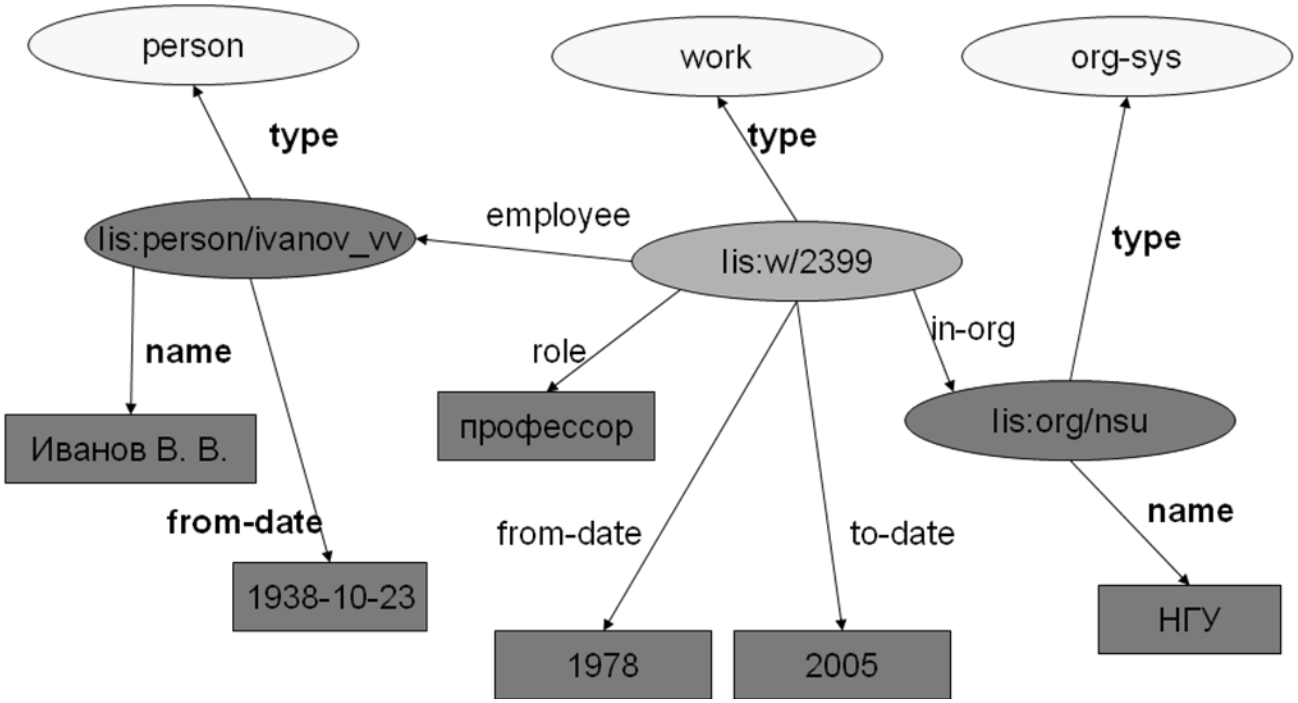


Рис. 3. Фрагмент данных, описывающих персону, организацию
и отношение между ними и использующих псевдосущности

Ещё одной важной особенностью, хотя и прямо не касающейся онтологического построения, являются расширения RDF, ориентированные на редактирование распределённых данных. Суть проблемы в том, что в распределённой системе не все данные являются доступными редактирующему агенту, то есть, «чужие» данные ему доступны для чтения, но не для изменения или уничтожения. Эту проблему в фактографических системах мы решаем следующим образом. Единицей редактирования определяется запись, введённая ранее по тексту. Это из-за того, что отдельное высказывание слишком «мало» и не имеет самостоятельной идентификации. Запись же такую идентификацию имеет. Полный набор редактирующих действий включает в себя добавление новой записи, уничтожение имеющейся записи и редактирование записи. Первое действие выполняется стандартным добавлением соответствующего фрагмента графа. Уничтожение записи стандартом RDF не предусмотрено. Для уничтожения записи введён оператор `delete`. Также стандартом RDF не предусмотрены конструкции, ведущие к изменению уже введённых полей и ссылок. В нашем подходе изменение записи выполняется через введение нового её варианта, имеющего ту же идентификацию `rdf:about`, но более позднюю временну́ю отметку. Этим способом можно изменять как свои, так и «чужие» записи.

## 2. Общая архитектура онтологии BONE

Базовая онтология неспецифических сущностей определяет следующие основные классы: персоны, организационные системы, географические системы, документы и отношения между ними. Кроме того, есть ряд дополнительных сущностных классов, таких как коллекции, архивы, предметы и др. На рис. 4 изображены основные сущностные классы и отношения между ними.

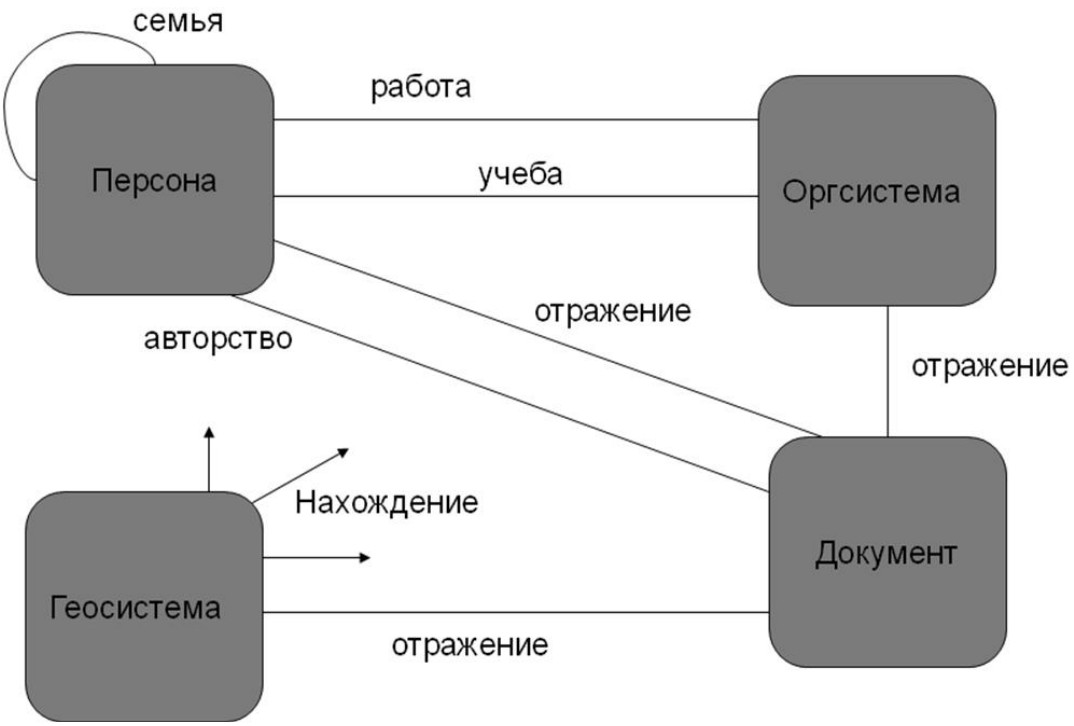


Рис. 4. Основные сущностные классы и отношения между ними

Как видно из рисунка, имеется 4 основных класса сущностей: персоны, организационные системы, документы и географические системы. Под организационными системами понимаются все формальные и неформальные объединения людей для достижения общих целей. В эту категорию попадают организации, туристические группы, клубы, конференции и т.д. и т.п. На рисунке дугами также указаны основные отношения между классами. Между персонами задаются родственные и семейные отношения, между организационными системами и людьми есть отношения типа «работа» и типа «учёба». Документы могут иметь авторов, а это задаёт отношение авторства, кроме того, документы отражают внешний мир. Например, на фотодокументе может быть изображён конкретный человек или группа людей. Географические системы через отношение «нахождение» являются местами расположения объектов других классов. Всё это только условная схема, более точно и подробно онтология будет охарактеризована далее по тексту.

## 3. Некоторые особенности формальной спецификации и средств формальной спецификации

Идейно, онтология была сформирована несколько лет назад, тогда же появился вариант формальной спецификации на языке OWL. В дальнейшем в онтологию и её спецификацию вносились некоторые добавления и изменения. Это отражало эволюционный процесс перехода к новым задачам и не приводило к проблемам с обрабатываемыми данными. Однако накопившаяся группа идей, а также потребность в выходе в мировые пространства имён и данных, привели к достаточно существенной переработке онтологии.

Новый вариант идейно очень близок к «старому», но внесены изменения в следующих направлениях:

- введены «наши» пространства имён – одно для расширения средств RDFS и OWL, другое – пространство имён проектируемой онтологии;
- изменён стиль формирования онтологических идентификаторов. Ранее, идентификаторы состояли из маленьких букв, а роль разделителя играл знак «минус». Теперь

разделитель реализуется через большие буквы каждого следующего в идентификаторе слова, имена классов начинаются с большой буквы, идентификаторы свойств – с маленькой;
- мы отказались от активного использования иерархии классов, вместо этого, где можно, использованы перечислимые классификаторы. Например, есть класс `Document`, и у него есть перечислимое поле `DocumentClassification`, в вариантах значений которого есть и тексты, и фотографии, видео и аудио;
- описания приведены в соответствие со стандартом OWL.

Для формальной спецификации онтологии средств, рекомендуемых в OWL/RDFS, оказалось недостаточно, поэтому мы расширили язык описания онтологий некоторыми дополнительными элементами.

1. В некоторых случаях желательно указать то, что определяемый класс является абстрактным, т.е. экземпляров элементов этого класса не должно появляться. Введённая конструкция выглядит следующим образом:

```
<owl:Class rdf:about="&fog;SystemObject" isabstract="true">
  <rdfs:subClassOf rdf:resource="&fog;Entity"/>
</owl:Class>
```

2. Техника определения перечислимых значений, рекомендованная в OWL, выглядит неудобной и ненаглядной. Использовался способ явного и компактного определения набора перечисления. Это выглядит следующим образом:

```
<EnumerationType rdf:about="&fog;PersonGenderEnum">
  <state value="m" xml:lang="ru">муж.</state>
  <state value="f" xml:lang="ru">жен.</state>
  <state value="m" xml:lang="en">male</state>
  <state value="f" xml:lang="en">female</state>
</EnumerationType>
```

Логика такой определяющей конструкции достаточно очевидна.

3. Введение в RDFS меток `<rdfs:label>… </rdfs:label>` достаточно удобно и позволяет приложению вставлять «осмысленные» слова в интерфейсе. Однако для определения `<owl:ObjectProperty …>…</ owl:ObjectProperty>` одной метки не хватает. Это потому, что название отношения может быть разным с разных «сторон» его использования. Например, предположим, мы определяем прямое объектное свойство «отец» между персонами. И со стороны отца, и со стороны сына в их информационных портретах «противоположный» элемент этого свойства будет виден. Но в одном случае, поле будет называться «отец», а в другом случае – «ребёнок». Был введён элемент inverse-label, смысл его на приведённом примере можно пояснить фрагментом определения:

```
<owl:ObjectProperty rdf:about="father">
  <rdfs:label xml:lang="ru">отец</rdfs:label>
  <rdfs:label xml:lang="en">father</rdfs:label>
  <inverse-label xml:lang="ru">ребенок</inverse-label>
  <inverse-label xml:lang="en">child</inverse-label>
  <rdfs:domain rdf:resource="&fog;Person"/>
  <rdfs:range rdf:resource="&fog;Person"/>
</owl:ObjectProperty>
```

Введённые дополнения к средствам спецификации не противоречат RDF и OWL и также специфицированы в языке OWL. Заметим, что, по правилам RDF, одиночный `DatatypeProperty` элемент может изображаться в виде атрибута.

Также были определены некоторые специальные средства структуризации данных. Специальные в том смысле, что при общей RDF-корректности, они имеют специальную семан-

тику. Они используются платформенным решением для определённых целей. Все эти средства имеют отношение к редактированию RDF-данных. Редактирование включает в себя создание новых узлов и свойств, изменение или уничтожение имеющихся узлов и свойств. Создание новых узлов – единственный процесс, относящийся к редактированию, поддержанный имеющимися стандартами. Выполняется он, очевидно, добавлением новых узлов и свойств.

По остальным действиям создана следующая система соглашений. Квантом изменения или уничтожения, в нашем подходе, является запись. В терминах множества триплетов, записью является набор всех триплетов, в которых поле субъекта одинаково. В терминах XML-представления RDF-данных, это сведенный под единый элемент `rdf:Description` всех полей и объектных свойств, относящихся к одному идентификатору `rdf:about`. Запись может быть уничтожена или изменена. Уничтожение записи с идентификатором id производится либо внедрением поля `<delete/>` в набор полей записи, либо добавлением к общему набору данных отдельного элемента:

```
<rdf:Description rdf:about=”id”>
  <delete/>
</rdf:Description>
```

Изменение записи производится добавлением изменённого варианта записи, но с более поздней временно́й отметкой. Она представляет собой значение какого-то момента времени в шкале универсального времени (`UniversalTime`), и добавляется к записи для того, чтобы зафиксировать время последнего её изменения. Такая отметка реализуется либо полем (`DatatypeProperty`), напр.

```
<fog:Person rdf:about=”…”>
  <mT>2013-01-22 14:02:50Z</mT>
  …
</fog:Person>
```

либо атрибутом, что, по правилом RDF, является допустимым:

```
<fog:Person rdf:about=”…” mT=” 2013-01-22 14:02:50Z”>
  …
</fog:Person>
```

## 4. Обзор спецификации онтологии BONE

Онтологический словарь выстроен в пространстве имён (namespace) `http://fogid.net/2012/`, которому обычно сопоставляется префикс fog. Онтология имеет корневой класс `fog:Entity`, который в соответствии с принципами OWL наследуется от `http://www.w3.org/2002/07/owl#Thing`. Для сущностей (`fog:Entity`) определены свойства `fog:comment` – комментарий и временны́е отметки `fog:startDate`, `fog:endDate`.

Сущностный класс `fog:Entity` распадается на два абстрактных подкласса: `fog:SystemObject` – системный объект и `fog:RelationObject` – объектное отношение. Соответственно, `SystemObject` иерархически группирует собственно объекты (напр. упоминавшиеся «персона», «документ» и др.), а `RelationObject` – множество псевдосущностей, являющихся составными отношениями между системными объектами.

К классам системных объектов относятся:

- `fog:Person` – персона,
- `fog:OrganizationSystem` – организационная система,
- `fog:GeoSystem` – географическая система,
- `fog:Document` – документ,
- `fog:Archive` – архив,
- `fog:Collection` – коллекция,
- `fog:ThingSystem` – вещь.

Пример спецификации одного из классов:

```
<owl:Class rdf:about="&fog;Person">
  <rdfs:label xml:lang="ru">Персона</rdfs:label>
  <rdfs:label xml:lang="en">Person</rdfs:label>
  <rdfs:subClassOf rdf:resource="&fog;SystemObject"/>
</owl:Class>
```

Общими свойствами для системных объектов являются:

- `fog:name` – имя объекта,
- `fog:description` – определение, абстракт, аннотация и т.д.

Многие системные объекты неоднородны, напр. класс организационных систем состоит из категорий: организации, мероприятие, объединение, отдел и «другое». Такая категоризация производится не средствами подклассов и наследований, а внесением в данный класс свойства `Classificator`, перечисляющего варианты. Например:

```
<owl:DatatypeProperty rdf:about="&fog;OrganizationSystemClassification">
  <rdfs:label xml:lang="ru">классификация орг.системы</rdfs:label>
  <rdfs:label xml:lang="en">organization system classification</rdfs:label>
  <rdfs:domain rdf:resource="&fog;OrganizationSystem"/>
  <rdfs:range rdf:resource="&fog;OrganizationSystemClassificationEnum"/>
</owl:DatatypeProperty>
<EnumerationType rdf:about="&fog;OrganizationSystemClassificationEnum">
  <state value="organization" xml:lang="ru">организация</state>
  <state value="organization" xml:lang="en">organization</state>
  <state value="arrangement" xml:lang="ru">мероприятие</state>
  <state value="arrangement" xml:lang="en">event</state>
  <state value="union" xml:lang="ru">объединение</state>
  <state value="union" xml:lang="en">union</state>
  <state value="department" xml:lang="ru">отдел</state>
  <state value="department" xml:lang="en">department</state>
  <state value="other" xml:lang="ru">другое</state>
  <state value="other" xml:lang="en">other</state>
</EnumerationType>
```

Большинство отношений между системными объектами реализуется техникой псевдосущностей. То есть, создаётся класс, имеющий родителем `fog:RelationObject`, а в нём определяются одна-две ссылки (`ObjectProperty`) на участников отношения и нужный набор атрибутов отношения. Например, отношение между системным объектом и организационной системой называется «участие», фрагмент его спецификации выглядит следующим образом:

```
<owl:Class rdf:about="&fog;Participation">
  <rdfs:label xml:lang="ru">Участие</rdfs:label>
  <rdfs:label xml:lang="en">Participation</rdfs:label>
  <rdfs:subClassOf rdf:resource="&fog;RelationObject"/>
</owl:Class>
<owl:ObjectProperty rdf:about="&fog;participant">
```

```
      <rdfs:label xml:lang="ru">участник</rdfs:label>
      <rdfs:label xml:lang="en">participant</rdfs:label>
      <inverse-label xml:lang="ru">участник в орг.</inverse-label>
      <inverse-label xml:lang="en">participant in organization</inverse-label>
      <rdfs:domain rdf:resource="&fog;Participation"/>
      <rdfs:range rdf:resource="&fog;SystemObject"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:about="&fog;participatesIn">
      <rdfs:label xml:lang="ru">в орг. сист.</rdfs:label>
      <rdfs:label xml:lang="en">in organization system</rdfs:label>
      <inverse-label xml:lang="ru">работник/участник</inverse-label>
      <inverse-label xml:lang="en">worker/participant</inverse-label>
      <rdfs:domain rdf:resource="&fog;Participation"/>
      <rdfs:range rdf:resource="&fog;OrganizationSystem"/>
    </owl:ObjectProperty>
    <owl:DatatypeProperty rdf:about="&fog;role">
      <rdfs:label xml:lang="ru">роль</rdfs:label>
      <rdfs:label xml:lang="en">role</rdfs:label>
      <rdfs:domain rdf:resource="&fog;Participation"/>
      <rdfs:range rdf:resource="&fog;languagedText"/>
    </owl:DatatypeProperty>
```

Имеется два специфический унарных отношения: `fog:Naming` и `fog:Dating`. Вообще-то именование и датирование, в основном, производится свойствами `fog:name`, `fog:startDate`, fog:`endDate`. Но в ряде случаев этого недостаточно. Например, при наличии у объекта других названий (девичья фамилия и пр.) или при специфическом или неопределённом датировании. Введение таких отношений позволяет решить данные затруднения. Спецификация именования и датирования следующая:

```
<owl:Class rdf:about="&fog;Naming">
  <rdfs:label xml:lang="ru">Именование</rdfs:label>
  <rdfs:label xml:lang="en">Naming</rdfs:label>
  <rdfs:subClassOf rdf:resource="&fog;RelationObject"/>
</owl:Class>
<owl:ObjectProperty rdf:about="&fog;namedObject">
  <rdfs:label xml:lang="ru">именуемое</rdfs:label>
  <rdfs:label xml:lang="en">by name</rdfs:label>
  <inverse-label xml:lang="ru">именуется</inverse-label>
  <inverse-label xml:lang="en">being called</inverse-label>
  <rdfs:domain rdf:resource="&fog;Naming"/>
  <rdfs:range rdf:resource="&fog;SystemObject"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:about="&fog;namedAlias">
  <rdfs:label xml:lang="ru">друг. имя</rdfs:label>
  <rdfs:label xml:lang="en">another name</rdfs:label>
  <rdfs:domain rdf:resource="&fog;Naming"/>
  <rdfs:range rdf:resource="&fog;LanguagedString"/>
</owl:DatatypeProperty>
<owl:Class rdf:about="&fog;Dating">
  <rdfs:label xml:lang="ru">Датирование</rdfs:label>
  <rdfs:label xml:lang="en">Dating</rdfs:label>
  <rdfs:subClassOf rdf:resource="&fog;RelationObject"/>
</owl:Class>
<owl:ObjectProperty rdf:about="datingEntity">
  <rdfs:label xml:lang="ru">датируемое</rdfs:label>
  <rdfs:label xml:lang="en">by date</rdfs:label>
  <inverse-label xml:lang="ru">датируется</inverse-label>
  <inverse-label xml:lang="en">is dated</inverse-label>
  <rdfs:domain rdf:resource="&fog;Dating"/>
  <rdfs:range rdf:resource="&fog;Entity"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:about="&fog;specificDate">
```

```
  <rdfs:label xml:lang="ru">дата</rdfs:label>
  <rdfs:label xml:lang="en">date</rdfs:label>
  <rdfs:domain rdf:resource="&fog;Dating"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="&fog;datingAccuracy">
  <rdfs:label xml:lang="ru">точность датирования</rdfs:label>
  <rdfs:label xml:lang="en">dating accuracy</rdfs:label>
  <rdfs:domain rdf:resource="&fog;Dating"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
</owl:DatatypeProperty>
```

Всего в онтологии 25 классов, 28 атрибутов (`DatatypeProperty`), 27 ссылок (`ObjectProperty`).

## 5. Использование онтологии BONE

Онтология BONE создавалась в течение более 10 лет. При этом на всех этапах она использовалась для построения различных баз данных, созданных в ИСИ СО РАН. К наиболее важным базам данных относятся: база данных Фотоархива СО РАН, база данных Открытого архива СО РАН, база данных Школы юных программистов, база данных выпускников и преподавателей ММФ НГУ. Интерфейсы к соответствующим информационным системам размещены в сети Интернет и доступны по адресам: `http://soran1957.ru`, `http://odasib.ru`, `http://mag.iis.nsk.su/syp/`, `http://mmf50.ru`.

## Литература


1. *Марчук А.Г.* Распределённые электронные архивы, библиотеки и базы данных. Препринт 122, Институт систем информатики им. А.П. Ершова СО РАН, Новосибирск, 25с., 2004.
2. *Alexander Marchuk* «Methods and Technologies of Digital Historical Factography», in Knowledge Processing and Data Analysis. First International Conference, KONT 2007, Novosibirsk, Russia, September 14-16, 2007,and First International Conference, KPP 2007, Darmstadt, Germany, September 28-30, 2007. Revised Selected Papers. Series: Lecture Notes in Computer Science, Vol. 6581. Subseries: Lecture Notes in Artificial Intelligence. Wolff, K.E.; Palchunov, D.E.; Zagoruiko, N.G.; Andelfinger, U. (Eds.) 2011, ISBN 978-3-642-22139-2.
3. *Марчук А.Г., Марчук П.А.* Платформа интеграции электронных архивов // Электронные библиотеки: перспективные методы и технологии, электронные коллекции. Труды девятой всероссийской конференции. Переславль, 2007, с. 89-94.
4. *Berners-Lee Tim, Hendler James, Lassila Ora* «The Semantic Web», in Scientific American, volume 284(5), 2001, pages 34-43.
5. Resource Description Framework (RDF). URL: `http://www.w3.org/RDF` (дата обращения 20.08.2014)
6. Web Ontology Language (OWL). URL: `http://www.w3.org/2004/OWL` (дата обращения 20.08.2014)
7. *Марчук А.Г., Марчук П.А.* Особенности построения цифровых библиотек со связанным контентом // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Сб.трудов / XII Всеросс. научн. Конф. RCDL'2010, Казань, Россия 13-17 октября 2010 г. - Казань: Казан. ун-т, 2010. - С. 19-23.


*Статья поступила в редакцию 01.09.2014.*

**Марчук Александр Гурьевич**
д.ф.-м.н., профессор, директор Института систем информатики им. А.П. Ершова Сибирского отделения РАН, (630090, Новосибирск, пр. Ак. Лаврентьева, д.6), тел. (383) 3308652, e-mail: mag@iis.nsk.su.

**Марчук Петр Александрович**
м.н.с. Института систем информатики им. А.П. Ершова Сибирского отделения РАН, (630090, Новосибирск, пр. Ак. Лаврентьева, д.6), тел. (383)3307068,
e-mail: peter@iis.nsk.su.

**Basic ontology of nonspecific entities BONE and its use for information systems construction**

**A.G. Marchuk, P.A. Marchuk**

The paper discusses the principles of basic ontology of nonspecific entities BONE, some features of its specification and use of this ontology while creating databases and information systems. A distinctive feature of the presented ontology is the systematic use of so-called constituent relations. It allows you to define attributes not only for objects, but also for relationships between them. Ontology BONE is intended for information structuring within the bounds of the approaches and technologies of Semantic Web.

*Keywords*: BONE, ontology, Semantic Web, information systems.