

УДК 004.8

### ПРИНЦИПЫ ИНТЕГРАЦИИ СОДЕРЖИМОГО RDF-ХРАНИЛИЩ В ПРОЕКТ OSTIS

Каешко А.И.\* , Колб Д.Г.\*\*

\* *Институт пограничной службы Республики Беларусь,  
г. Минск, Республика Беларусь*  
**ondister@gmail.com**

\*\* *Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*  
**kolb@bsuir.by**

Рассмотрена общая идея трансляции содержимого RDF-хранилищ в семантические сети с базовой теоретико-множественной интерпретацией. Описаны основные этапы трансляции, подробно разобран один из этапов.

**Ключевые слова:** транслятор, N-triples, Semantic Web, RDF, SC-код, онтологии

#### Введение

Высокие темпы развития информационных технологий в области онтологий и онтологического инжиниринга сопровождаются развитием различных направлений, форм и средств представления онтологий [Загорюлько, 2011].

Для многих предметных областей уже созданы обширные онтологии, такие как SNOMED в медицине, инженерные онтологии PhysSys, EngMath, существуют и межпредметные онтологии, такие как NAICS или SUMO. Стоит отметить, что большинство онтологических ресурсов являются англоязычными. Некоторые онтологии являются международными, и активно используются, например международная классификация болезней 10 пересмотра (МКБ-10).

Наиболее популярным направлением, в рамках которого разрабатываются онтологии верхнего уровня, онтологии предметных областей и прикладные онтологии [Соловьев, 2006], является Semantic Web. С момента рождения направления в 2001 году создано огромное количество онтологий для различных предметных областей.

Одной из ключевых задач развития проекта мы считаем интеграцию уже накопленных другими разработчиками знаний в рамках баз знаний систем, построенных на базе технологии OSTIS.

Основными проблемами, стоящими на пути к решению данной задачи являются:

- разнородность систем метаданных,

использующихся в различных онтологиях;

- особенности описания предметной области присущие конкретному инструменту, к примеру, использование только бинарных отношений (OWL, RDF).

#### 1. Системы метаданных.

С ростом количества информационных ресурсов, поддерживающих стандарты Semantic Web, появилась необходимость унификации представления хранящихся в них знаний. Решением этой проблемы стало использование для представления знаний в информационных ресурсах систем метаданных. В настоящее время существует несколько десятков проектов, связанных с разработкой систем метаданных [Антопольский, 2012]. Одним из наиболее популярных проектов, направленных на решение проблемы унификации представления знаний в виде семантических сетей, стал проект «Дублинское ядро» [DC, 2012]. Основными результатами проекта являются словари метаданных общего назначения, стандартизирующие описание ресурсов с помощью различных RDF-форматов.

Анализируя работы по развитию метаданных можно сделать вывод о том, что основными проблемами, связанными с использованием систем метаданных, являются:

- абстрагируемость большинства систем метаданных от технологий представления и обработки информации, что не позволяет формировать метаописания всего информационного

ресурса и приводит к проблемам “глубинного web” [Ландэ,2009];

– ориентация большинства систем метаданных на создание метаописаний на каком-то одном уровне представления информации, что в итоге создает проблемы при описании самого ресурса и его фрагментов или совокупности ресурсов в рамках одной системы метаданных;

– наличие систем метаданных, предназначенных для решения одной задачи, но построенных на разных понятийных базисах, которые могут частично пересекаться или не пересекаться вовсе. Ярким примером может служить универсальные системы метаданных GILS [GILS, 2012] и Dublin Core, в которых ключевые элементы пересекаются.

Таким образом, назревает необходимость использования инструментов, которые могли интегрировать и унифицировать существующее многообразие систем метаданных. Наиболее перспективным в этом отношении нам представляются семантические сети с базовой теоретико-множественной интерпретацией, развиваемые в проекте OSTIS.

С точки зрения проекта OSTIS система метаданных определяется как язык семантических сетей, имеющий определенную семантику в рамках некоторой предметной области или класса предметных областей.

Каждому классу предметных областей с заданной сигнатурой можно поставить в соответствие язык семантических сетей с заданным алфавитом и набором ключевых узлов [Голенков, 2011]. Каждому из языков семантических сетей можно поставить в соответствие множество конструкций SC-кода, в которых, кроме ключевых узлов самого SC-кода используются ключевые узлы, определяемые сигнатурными элементами соответствующего класса предметных областей. Каждое такое множество конструкций SC-кода будем называть sc-языком, ориентированным на представление структурных моделей предметных областей соответствующего класса.

Определим sc-язык, как язык семантических сетей, ориентированный на представление структурных моделей предметных областей определенного класса.

Каждый sc-язык задается алфавитом (который совпадает с алфавитом SC-кода) и семейством ключевых узлов, которое включает в себя семейство ключевых узлов SC-кода.

Таким образом, процесс интеграции систем метаданных в проект OSTIS заключается в определении SC-языка или SC-языков, ключевыми узлами, которых являются выделенные в рамках систем метаданных понятия.

В настоящий момент в рамках проекта OSTIS выделено пятнадцать SC-языков, которые постоянно дорабатываются и совершенствуются на основе ключевых элементов существующего теоретико-множественного аппарата и универсальных систем метаданных, разрабатываемых мировым научным сообществом.

## 2. Инструменты представления онтологий и их свойства.

В настоящее время определено несколько крупных научных проектов, в которых сформировался инструментарий, позволяющий создавать семантическое описание различных предметных областей. В первую очередь здесь необходимо отметить проект OpenCyc [Лапшин,2010] – старейшее хранилище онтологий со своим уникальным инструментарием, проект schema.org от крупнейших поисковых гигантов, определяющий свой инструмент для разметки web-ресурсов (microdata) и проект Semantic web, который на сегодняшний день является “законодателем моды” среди онтологического инструментария, на который ровняется большинство проектов.

Несомненным плюсом проекта Semantic web является ориентация на независимую распределенную разработку онтологий. Знания о предметной области могут накапливаться и уточняться постепенно, с участием большого числа людей, без постоянного согласования [Трофимов, 2011].

Однако инструменты Semantic web не дают ответа на вопрос, как избежать добавления в онтологию противоречивых утверждений и что делать, если противоречия возникнут.

Наиболее обсуждаемыми проблемами инструментария Semantic web и в частности языка OWL являются:

- что моделировать при помощи классов, а что при помощи экземпляров. В инструментах Semantic web существует неоднозначность при определении классов и их экземпляров;
- еще одним недостатком является отсутствие возможности естественным образом определять свойства у свойств. Это не позволяет моделировать атрибуты у предметных отношений, n-арные отношения и атрибуты у атрибутов;
- ориентация проекта на web и близкое к машинному представление семантических сетей;
- не развиты стандарты представления переменных во времени и нечетких предметных областей;
- слабо проработанный уровень верификации онтологий на противоречивость и полноту.

Очевидно, что, не смотря на свою популярность, инструменты Semantic web имеют ряд недостатков. В первую очередь они связаны с тем, что инструменты Semantic web изначально были нацелены на машиноориентированное описание информационных ресурсов в web-пространстве без учета комплексного решения проблем семантического представления с точки зрения теории искусственного интеллекта.

В противоположность инструментам Semantic web инструменты проекта OSTIS имеют строгую теоретико-множественную трактовку и не привязаны к конкретной прикладной области. Что

обеспечивает более компактное и формально точное представления информации [Голенков, 2011].

Это определяется рядом свойств, которые позволяют говорить о языковых средствах проекта OSTIS, как наиболее предпочтительном средстве интеграции знаний из различных источников:

- как и в языках Semantic web в технологии OSTIS отдается предпочтение бинарным отношениям, однако существует возможность представления отношений любой арности;
- отношения представляются в виде узлов семантической сети, что позволяет характеризовать их свойства;
- экземпляры отношений выделяются как отдельные узлы семантической сети, что дает возможность характеризовать каждый экземпляр отношения уникальным образом;
- в алфавите ключевых узлов и дуг имеются элементы для описания нечетких, негативных и нестационарных объектов.

### 3. Этапы трансляции содержимого RDF-хранилищ в тексты SC-кода

На основании частей 1 и 2 данной работы представим общую идею трансляции содержимого RDF-хранилищ в тексты SC-кода.

**Предварительный этап:** Преобразование наиболее популярных систем метаданных в sc-языки. В рамках такого преобразования учитывается, что различные системы метаданных могут пересекаться по ключевым элементам. Само преобразование осуществляется вручную, ввиду важности такого преобразования, и наличия, как правило, небольшого количества ключевых элементов в рамках систем метаданных. Пересекающиеся элементы различных систем метаданных оформляются одинаково.

**Этап 1:** Определение использования известных транслятору систем метаданных в рамках RDF-хранилища. Согласно документации Semantic web, используемая система метаданных оформляется в виде RDF-словаря и помещается в начале определения любого RDF-документа. В рамках этапа возможна верификация RDF-документа на предмет корректного использования системы метаданных в данном документе. Такая верификация может быть осуществлена на основе схем отношений, представленных в определении RDF-словаря конкретной системы метаданных.

**Этап 2:** Трансляция с использованием существующих описаний систем метаданных. На данном этапе осуществляется преобразование той части содержимого RDF-хранилища, которая соответствует вывлеченным в RDF-хранилищах системам метаданных.

**Этап 3:** Трансляция нераспознанных фрагментов RDF-хранилища. На этом этапе части содержимого RDF-хранилища, которые не соответствуют известным транслятору системам метаданных, преобразуются предопределенную форму (для простоты в бинарные ориентированные отношения). Таких форм можно

определить несколько в зависимости от типа хранилища или типа языковых средств используемых в хранилище.

## 4. Трансляция нераспознанных объектов RDF-хранилища

### 4.1. Характеристики RDF и SC-кода

Язык RDF и SC-код имеют несколько общих черт. Во-первых, оба они могут быть представлены в виде графа, имеющего узлы и дуги. Во-вторых, все элементы такого графа должны иметь уникальные идентификаторы.

Атомарным объектом в RDF является триплет: СУБЪЕКТ – ПРЕДИКАТ – ОБЪЕКТ (рисунок 1).

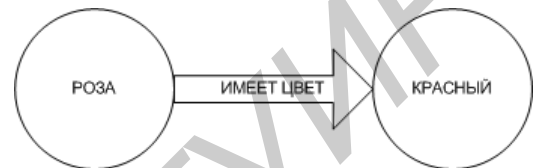


Рисунок 1 – Триплет RDF. «РОЗА», «ИМЕЕТ ЦВЕТ», «КРАСНЫЙ» – уникальные идентификаторы.

Субъект и объект являются узлами, а предикат направленной дугой, всегда указывающей от субъекта к объекту [RDF Concepts, 2013]. Уникальность идентификаторов должны обеспечивать международные идентификаторы ресурсов (IRI, international resource identifier) [RDF Semantics, 2013]. Считается, что посредством таких триплетов можно описать любой объект и отношения между этими объектами. Таким образом, множество триплетов образуют семантическую сеть. Для ограничения количества предикатов и исключения их противоречивости и дублирования используются онтологии верхнего уровня (Дублинское ядро), словарь RDFS, и описательную логику OWL. Однако технически ничто не мешает использовать произвольные предикаты в пространстве имен документа [RDF Freebase, 2013].

Простейшая трансляция представленного выше RDF графа в SC-код осуществляется с помощью введения неролевого отношения (рисунок 2).

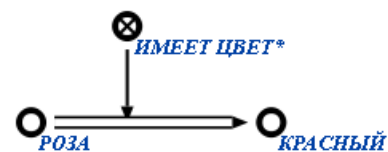


Рисунок 2 – Результат трансляции, представленный в виде SCg-конструкции.

Безусловно, трансляцию необходимо осуществлять между строковыми (линейными) представлениями RDF и SC-кода. При этом семантика RDF графа должна быть сохранена максимально. Для RDF форматом, наиболее удобным для понимания человеком и одновременно легким для машинного чтения, является N-triples, для SC-кода – SCs-код. В N-triples можно транслировать документ любого формата, выполненный на основе модели RDF.

Спецификация формата N-triples рекомендуется консорциумом Всемирной паутины (W3C) [RDF N-Triples, 2013]. Рекомендации от 5 ноября 2013 года предполагают, что каждый триплет содержится в одной строке и состоит из субъекта, предиката и объекта, разделенными символом табуляции или пробела. Каждая строка с триплетом заканчивается точкой. Субъект может быть представлен IRI ссылкой или пустым узлом (blank node), предикат всегда IRI ссылка, а объект может быть представлен IRI ссылкой, пустым узлом или литералом.

Как уже указывалось выше, IRI ссылки предназначены для обеспечения уникальной идентификации объектов. В N-triples они помещаются между символами «<» и «>». Например, `<http://example.org/#РОЗА>`.

Литералы используются для идентификации значений, таких как текст, даты, числа. Литералы заключаются в кавычки. Они подразделяются на литералы, относящиеся к определенному языку, литералы, имеющие тип данных, и простые литералы. Язык литерала указывается после символа «@», например, “Это литерал на русском языке”@ru. Тип данных литерала указывается после знаков «^^», и является IRI, например `"2"^^<http://www.w3.org/2001/XMLSchema#integer>`. Простые литералы не имеют явного указания типа, но, по сути, имеют тип данных `<http://www.w3.org/2001/XMLSchema#string>`.

Пустые узлы всегда начинаются с символов «\_:» и должны иметь уникальные идентификаторы в пределах документа. Пустые узлы стоит отметить отдельно. В RDF они могут использоваться для обобщенных утверждений. Например, триплет `<РОЗА> <ИМЕЕТ ЦВЕТ> _:x`, означает, что роза имеет какой-то цвет, но при этом не уточняет, какой именно. И наоборот, триплет `_:x <ИМЕЕТ ЦВЕТ> <КРАСНЫЙ>`, указывает, что какой-то объект имеет красный цвет. Пустые узлы так же используются для описания реальных объектов классов. Описание знакомств пользователя «Алексей» может выглядеть так:

```
_:bn1 <ОБЪЕКТ_КЛАССА> <ЧЕЛОВЕК>
_:bn1 <ИМЯ> "Алексей"
_:bn1 <ДАТА_РОЖДЕНИЯ>
"30.01.1979"^^<ДЕНЬ_МЕСЯЦ_ГОД>
_:bn1 <ЗНАЕТ> _:bn2
_:bn1 <ЗНАЕТ> _:bn3
_:bn2 <ИМЯ> "Илья"
_:bn2 <ДАТА_РОЖДЕНИЯ>
"12.05.1981"^^<ДЕНЬ_МЕСЯЦ_ГОД>
_:bn3 <ИМЯ> "Мария"
_:bn3 <ДАТА_РОЖДЕНИЯ>
"01.01.1967"^^<ДЕНЬ_МЕСЯЦ_ГОД>
```

Таким образом, синтаксис N-triples довольно простой. Предикат всегда является IRI ссылкой и объясняет связь субъекта с объектом. Кроме того, разнообразие предикатов теоретически ограничено словарями и онтологиями представления.

## 4.2. Этапы трансляции RDF-графов в sc-конструкции

Транслятор RDF-графов в sc-конструкции должен:

1. обладать высокой скоростью работы;
2. быть платформенезависимым, то есть поддерживать все аппаратные архитектуры и операционные системы, поддерживаемые технологией OSTIS;
3. обеспечивать семантическую и логическую эквивалентность исходных RDF-графов и получаемых sc-конструкций;
4. быть автоматизированным.

Трансляция RDF-графов в sc-конструкции должна состоять из последовательных этапов:

1. Трансляция RDF-дампа в SCs-код. На этом этапе должен быть выполнен перевод текстов формата N-triples в тексты SCs-кода по заранее определенным правилам трансляции. Данный этап должен быть максимально быстрым и автоматизированным. Реализация транслятора будет зависеть от платформы, на котором он будет выполняться, поэтому алгоритмы его работы должны быть максимально простыми и легко воспроизводимыми в трансляторах для различных платформ и операционных систем в будущем.

2. Трансляция SCs-кода в базу знаний в виде нового компонента. Выполнение этапа будет осуществляться с использованием уже разработанных инструментов технологии OSTIS. Платформенная независимость таких инструментов обеспечивается и поддерживается командой разработчиков OSTIS. На этом этапе необходим контроль архитектора информационной системы, так как ошибки, допущенные при трансляции на предыдущем этапе, становятся видны. Как правило, это неправильная или некорректная идентификация узлов. Однако может встречаться и некорректное для SC-кода определение отношений.

3. Интеграция нового компонента с существующей базой знаний, с использованием метасистемы OSTIS. На данном этапе происходит оптимизация структуры sc-конструкций, согласование понятий новой базы знаний с уже существующими.

## 4.3. Подход к трансляции N-triples в SCs-код

Документы, представленные на языке N-triples, часто имеют большие размеры. Дамп freebase, например, имеет размер более 100 гигабайт, и хранится в архиве Gzip, который позволяет построчное считывание файла без его полной распаковки. По причине больших объемов файлов трансляция из N-triples в SCs-код должна быть однопроходной. На языке C++ (компилятор MinGW) был создан прототип транслятора (далее транслятор). Разберем основные реализованные принципы его работы.

По аналогии с рисунками 1 и 2 триплет N-triples, состоящий только из IRI

```
<http://kaiko.getalp.org/kaiko/ontology/colors.owl>
```

```
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.w3.org/2002/07/owl#Ontology> .
```

при трансляции в scs-предложение будет выглядеть следующим образом:

```
colors => rdfs_type : w3owl_Ontology;;
```

Как видно из примера, у IRI выделяется пространство имен, а все символы, кроме цифр, букв заменяются символом «\_». Так, название онтологии

```
<http://kaiko.getalp.org/kaiko/ontology/colors.owl>
```

заменяется идентификатором «colors», а у предиката пространство имен

```
<http://www.w3.org/1999/02/22-rdf-syntax-ns>
```

заменяется префиксом «rdfs». Эти видоизмененные IRI будут являться системными идентификаторами в базе знаний OSTIS. Субъект и объект транслируются в sc-узлы, а предикат должен быть нерольевым отношением. Поэтому, транслятор создает файл \_norole\_rdfs\_type.scs следующего содержания:

```
rdfs_type <- sc_node_norole_relation;;
rdf_predicates -> rdfs_type;;
rdfs_type => nrel_main_idtf: [rdfs_type*_type the ru id
by manual*] (* <- lang_ru;; *);;
rdfs_type => nrel_main_idtf: [rdfs_type*]( * <- lang_en;;
*);;
```

То есть, каждый новый предикат включается во множество нерольевых отношений и RDF предикатов, так же для него определяются глобальный русскоязычный и англоязычный идентификатор. Русскоязычный идентификатор предиката необходимо отредактировать вручную, задав значение на русском языке. Это неприятная особенность, но, даже для онтологии SUMO, содержащей более чем 500000 триплетов и использующей собственный словарь предикатов, их число не превышает 70. В примере выше отредактированное scs-предложение с русским идентификатором будет выглядеть так:

```
rdfs_type => nrel_main_idtf: [Объект класса*] (* <-
lang_ru;; *);;
```

И на SCn-коде транслированное в базу знаний scs-предложение

```
colors => rdfs_type : w3owl_Ontology;;
```

будет иметь вид:

```
colors
=> Объект класса*:
w3owl_Ontology
```

SCs-код имеет аналоги пустых узлов N-triples. Они могут быть локальными (уникальными в пределах одного SCs документа) и глобальными (уникальными в пределах одной транзакции трансляции SCs-кода в базу знаний). Идентификаторы таких узлов не используются как системные и начинаются с символов «\_». Трансляция идентификатора пустого узла N-triples в пустой узел Scs-кода заключается в замене символов «\_» на символы «.».

При трансляции литералов типы данных и идентификаторы языков относятся к множеству

типов литералов и определяются как узлы, не являющиеся отношением:

```
numeric <- sc_node_not_relation;;
lit_types -> numeric;;
numeric => nrel_main_idtf: [numeric_type the ru id by
manual] (* <- lang_ru;; *);;
numeric => nrel_main_idtf: [numeric]( * <- lang_en;;
*);;
```

Они так же, как и предикаты имеют описание в отдельном файле и число таких файлов, как правило, не большое. Значение литерала при трансляции помещается в scs-рамку. Триплет

```
<M83_3> <has_Description>
"Osteomalacia"^^<w3xml#string>
```

после трансляции будет иметь вид (рис. 3):

```
icd_10_M83_3 => icd_10_has_Description :
[Osteomalacia] (* <- w3xml_string;; *);;
```



Рисунок 3 – SCg-код результата трансляции триплета с типизированным литералом.

Таким образом, технологически трансляция N-triples в SCs-код особых проблем не вызывает, однако ряд особенностей имеется.

Технология Semantic Web ориентирована в первую очередь на использование во Всемирной паутине. По этой причине часто IRI, содержат ссылку на RDF документ (то есть являются URL), описывающий данный IRI. Например, IRI <http://www.rdfabout.com/rdf/usgov/geo/us/ny> является идентификатором штата Нью-Йорк в Соединённых Штатах. Если перейти по указанному IRI, можно получить RDF-документ, описывающий Нью-Йорк. Такой подход, безусловно, хорош в сети, но в режиме оффлайн, при трансляции документа, такие IRI не будут иметь описание.

Определенной проблемой являются особенности синтаксиса N-triples в различных онтологиях и дампах. Так, идентификаторы пустых узлов могут быть заключены в символы «<>» и «>», не смотря на то, что этот подход не описан в рекомендациях W3C. Часто в литералах используются символы форматирования текста, например символ переноса строки. Учет всех этих допущений влияет в основном на скорость работы транслятора.

Ничем не ограниченное, кроме рекомендаций, множество предикатов и типов литералов создает трудности для автоматического согласования метасистемы OSTIS и онтологий представления RDF. Такое согласование необходимо производить вручную на следующих этапах трансляции.

Если говорить не только о фактах, представленных в виде триплетов, то стоит упомянуть RIF (Rule Interchange Format). Онтологии, построенные с использованием модели RDF, имеют богатые возможности описания знаний, но обладают весьма ограниченными средствами вывода следствий из имеющихся знаний. В RDF все

знания должны храниться явно в виде триплетов (фактов, аксиом). RIF является языком правил, и создан для вывода следствий, однако он имеет ограниченные описательные возможности. Комбинация RDF и RIF позволяет использовать достоинства обоих и устранить их недостатки. Таким образом, для получения новых триплетов RDF используются документ с правилами RIF и факты RDF. Использование RIF не обязательно, если онтология изначально наиболее полная, более того перед трансляцией можно получить все новые факты на основе правил, то есть пополнить онтологию.

## Заключение

В настоящее время практически реализована трансляция нераспознанных RDF-графов в сконструкции. Оптимизация алгоритмов трансляции будет продолжена, но уже сейчас можно говорить о реально существующей совместимости технологий Semantic Web и OSTIS на уровне линейных форматов представления знаний.

## Библиографический список

- [Антопольский, 2012] Антопольский, А.Б. Исследование и разработка системы метаданных для электронных информационных ресурсов и сервисов в фундаментальной науке / А.Б. Антопольский, В.И. Ауссем, С. А. Блау, А.И. Жежель // Отчет о результатах работ по гранту РФФИ № 04-07-90087 [Электронный ресурс]. – 2012. – Режим доступа: <http://rd.feb-web.ru/antopolsky-04.htm#4>. – Дата доступа: 3.04.2012.
- [Голенков, 2011] Голенков, В. В. Принципы построения массовой семантической технологии компонентного проектирования интеллектуальных систем / В.В. Голенков, Н. А. Гулякина // Материалы международной научно-технической конференции OSTIS-2011. – 2011. – Минск БГУИР. С. 21-58
- [Голенков, 2013] Голенков, В. В. Открытый проект, направленный на создание технологии компонентного проектирования интеллектуальных систем / В.В. Голенков, Н. А. Гулякина // Материалы международной научно-технической конференции (OSTIS-2013). – Минск: БГУИР, 2013, С. 55-77.
- [Загорюлько, 2011] Загорюлько, Ю. А. Подход к построению интеллектуальных информационных систем на основе семантических сетей. – В кн. Междунар. научн.-техн. конф. «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2011). Материалы конф. [Минск, 10-12 февр. 2011 г.]. – Минск: БГУИР, 2011, с. 15-20.
- [Ландэ, 2009] Ландэ, Д. В. Глубинный веб-информационная среда для бизнес аналитика / Д. В. Ландэ // Информационные технологии для менеджмента, 2009, № 9, с. 28-32
- [Лапшин, 2010] Лапшин, В.А. Система Сус и её библиотека онтологий / В. А. Лапшин // Искусственный интеллект и принятие решений, 2010, № 2, с. 42-53
- [Соловьев, 2006] Соловьев, В.Д. Онтологии и тезаурусы / уч. пособие [Электронный ресурс], Режим доступа: <http://bookre.org/reader?file=786973&pg=0> // Д.В. Соловьев, Б. В. Добров, В.В. Иванов, Н.В. Лукашевич. – Казань, Москва. – 2006. – 157 С.
- [Трофимов, 2011] Трофимов, И. В. Эволюция выразительных способностей языка OWL / И. В. Трофимов // Программные системы: теория и приложения : электрон. научн. журн. 2011. № 4(8), с. 85-94. URL: [http://psta.psiras.ru/read/psta2011\\_4\\_85-94.pdf](http://psta.psiras.ru/read/psta2011_4_85-94.pdf)
- [DC, 2012] The Dublin Core Metadata Initiative [Электронный ресурс]. – 2012. – Режим доступа: <http://dublincore.org/>. – Дата доступа: 30.04.2012
- [GILS, 2012] Global Information Locator Service (GILS). [Электронный ресурс]. – 2012. Режим доступа: <http://www.gils.net>. – Дата доступа: 3.04.2012.
- [GRDDL, 2007] Gleaning Resource Descriptions from Dialects of Languages (GRDDL). W3C Recommendation 11 September 2007 [Electronic resource]. – 2007. - URL: <http://www.w3.org/TR/grddl> – Дата доступа: 28.11.2013.
- [RDF Concepts, 2013] RDF 1.1 Concepts and Abstract Syntax. W3C Candidate Recommendation 05 November 2013 [Electronic

resource]. – 2013. - URL: <http://www.w3.org/TR/2013/CR-rdf11-concepts-20131105> – Дата доступа: 19.11.2013.

[RDF N-Triples, 2013] RDF 1.1 N-Triples. A line-based syntax for an RDF graph. W3C Candidate Recommendation 05 November 2013 [Electronic resource]. – 2013. - URL: <http://www.w3.org/TR/2013/CR-n-triples-20131105> – Дата доступа: 18.11.2013.

[RDF Freebase, 2013] RDF Overview. Freebase API. [Electronic resource]. – 2013. - URL: <https://developers.google.com/freebase/v1/rdf-overview> – Дата доступа: 18.11.2013.

[RDF Semantics, 2013] RDF 1.1 Semantics. W3C Candidate Recommendation 05 November 2013 [Electronic resource]. – 2013. - URL: <http://www.w3.org/TR/2013/CR-rdf11-mt-20131105> – Дата доступа: 18.11.2013.

## BASIC PRINCIPLES OF INTEGRATION OF RDF-STORAGE CONTENT INTO PROJECT OSTIS

Kayeshko A.I.\* , Kolb D.G.\*\*

*\*Institute of the border service of the Republic of Belarus, Minsk, Republic of Belarus*

*ondister@gmail.com*

*\*\*Belorussian state university of informatics and radioelectronics, Minsk, Republic of Belarus*

*kolb@bsuir.by*

In the article research the approaches and possibilities to translate N-Triples format of the RDF-model into SCs-code. Implementation of the possibility a RDF to SCs-code translation will reduce the time of the filling an OSTIS knowledge storages by a Semantic Web technology content.

## Introduction

To ensure semantic interoperability between intelligent information systems use ontologies. One of the most promising technologies of intelligent information systems is the Open Semantic Technologies for Intelligent Systems (OSTIS). For fast filling OSTIS knowledge storages, necessary to use an already created ontology.

## Main Part

Ontologies created on the basis of model RDF, ubiquitous. The format of the RDF-model, most convenient for human understanding and at the same time the machine-readable, is N-triples, for SC-code such format is SCs-code. All RDF-model based documents can be translated in N-triples document.

Due to the large size of some RDF-dumps, selected single-pass translation algorithm. During the process of translation separated and described in separate files predicates and literals defining entity classes. Due to an unlimited number of predicates and literals it is difficult to create the automatic translator.

## Conclusion

The proposed approach to translation RDF N-triples in SCs-code is fairly primitive, mainly due to a single-pass translation algorithm. Translator optimization will continue, but now we can talk about real compatibility between Semantic Web technologies and OSTIS at linear format of knowledge representation.