

УДК 004.048

КОНВЕРТЕР ОНТОЛОГИЧЕСКОЙ БАЗЫ ЗНАНИЙ СТАНКОВ И ДЕТАЛЕЙ МАШИНОСТРОИТЕЛЬНОГО ПРОИЗВОДСТВА В ОБЪЕКТНУЮ БАЗУ ДАННЫХ НА ОСНОВЕ ANTLR

В.А. Игруша¹, С.С. Сосинская²

Иркутский национальный исследовательский технический университет, Иркутск, Россия

¹хоу-90@mail.ru, ²sosinskaya@mail.ru

Аннотация

Статья посвящена разработке автоматизированного конвертера, преобразующего базу знаний в виде онтологии, представленной в одном из форматов, допустимых редактором онтологий OntoStudio (OWL или RDF), в объектно-ориентированную базу данных (ООБД) db4o. Впервые предложен подход, основанный на использовании компилятора компиляторов ANTLR (ANother Tool for Language Recognition). Это позволяет сократить трудозатраты и применить программное средство ANTLR для других онтологий, так как разработана грамматика языка описания онтологий, которая не зависит от предметной области базы знаний. Грамматика содержит не только лексические и синтаксические правила для проверки правильности текста, но и правила действий для преобразования элементов онтологии в изоморфные им элементы объектной базы данных.

Ключевые слова: онтология, OntoStudio, ANTLR, Protégé, конвертер, ООБД, db4o, проект, база знаний, класс.

Цитирование: Игруша, В.А. Конвертер онтологической базы знаний станков и деталей машиностроительного производства в объектную базу данных на основе ANTRL / В.А. Игруша, С.С. Сосинская // Онтология проектирования. – 2016. – Т. 6, №3(21). - С. 278-286. – DOI: 10.18287/2223-9537-2016-6-3-278-286.

Введение

Существует множество обстоятельств, которые затрудняют распространение и обмен между людьми знаниями, столь необходимыми для их систематического пополнения и накопления.

Это связано с быстрым ростом численности населения и вовлечением новых поколений в различные сферы деятельности, требующие постоянно растущего уровня знаний, умений и навыков. Особым обстоятельством являются фундаментальные отличия и множественность национальных языков.

Многие проблемы в создании знаний и обмене ими обусловлены неоднозначным восприятием смысла знаний различными участниками процесса, которые могут пользоваться различной терминологией и понятийным аппаратом и руководствоваться различными моделями деятельности.

Вследствие этих объективных обстоятельств весьма желательно, чтобы информация и знания были структурированы таким образом, чтобы пользователь, в том числе и компьютер, понимал и текст, и смысл документа и обрабатывал его посредством использования известных ему правил с помощью некоторого логического языка, а также был способен вывести новые факты и знания из документа [1].

Онтология – один из способов структурирования знаний путём создания иерархии понятий с их свойствами, правилами и запросами. Понятия вместе с их экземплярами создают объектную базу знаний [2]. Правила позволяют выводить новые экземпляры [3].

С другой стороны, объектно-ориентированные базы данных (ООБД) представляют собой набор классов, образующих иерархию и содержащих свойства, методы, запросы и объекты (экземпляры) классов [4].

Информационная модель данных в области машиностроения требует перехода от неструктурированных данных к таким структурам, которые позволяют получить ответы на нужные запросы за наименьшее время. Для решения этой проблемы был предложен новый подход, сочетающий формирование онтологии, содержащей информацию о технологических возможностях станков и обрабатываемых на них деталей [5], и параллельно с этим проектирование и реализацию хранения той же информации в виде ООБД [6].

Разработка онтологии даёт возможность автоматизировать процедуру выбора станков, наиболее подходящих для обработки тех или иных деталей. Применение онтологии как гибкой системы знаний, в которую могут быть включены самые различные характеристики оборудования, позволяет создать технологический процесс, обеспечивающий минимальные затраты оборудования и снижение расходов на его техническое обслуживание.

Рассмотрение ООБД является существенно более эффективным подходом к разработке баз данных по сравнению с широко используемыми реляционными базами данных.

Разработка различных структур представления данных позволит в будущем сравнивать оба способа по скорости обработки запросов и объёму необходимой памяти и переходить от одного к другому для выбора наиболее эффективного в тот или иной период времени.

1 Цель разработки конвертера

Как показано в [6], онтология и ООБД для одной предметной области изоморфны.

Если представить онтологию $N = \{C, A, E, Q, R, P\}$, а объектно-ориентированную базу данных $V = \{C, S, O, Z, M, D\}$, где

C – классы;

A – атрибуты;

E – экземпляры классов;

Q – запросы;

R – правила;

P – отчеты;

S – свойства классов;

O – объекты, то есть экземпляры классов;

Z – результаты запросов;

M – методы;

D – отчетные документы,

то утверждается, что $N \cong V$, так как $A \cong S$, $E \cong O$, $Q \cong Z$, $R \cong M$, $P \cong D$.

В связи с этим необходимо иметь программное средство для преобразования из одной формы в другую.

Авторами было разработано приложение для сканирования онтологии [6], представленной в формате RDF-файла; этот формат описан в [7]. Такой файл является результатом совместной обработки редакторами онтологий OntoStudio и Protégé.

Результатом работы приложения является ООБД db4o.

Однако написание подобного приложения является очень трудоёмкой задачей и потребует больших затрат времени программиста при внесении хотя бы небольших

изменений в онтологию, а тем более при разработке расширенной онтологии, что является необходимым, например, при рассмотрении технологии изготовления деталей на станках.

На основании изложенного было принято решение разработать конвертер с использованием ANTLR (от ANOther Tool for Language Recognition) - инструментария для создания компилятора языков программирования, который не только проверяет правильность текста на входном языке с точки зрения лексики и синтаксиса, но и преобразует описание с языка описания онтологий в язык ООБД [8].

2 Состав и назначение ANTLR

ANTLR — генератор парсеров, позволяющий автоматически создавать программы лексического и синтаксического анализа на одном из целевых языков программирования (C++, Java, C#, Python, Ruby) по описанию LL - грамматики в расширенной нотации Бэкуса – Наура [8]. В общем случае грамматика относится к классу LL(K) грамматик, если для неё можно построить анализатор, учитывающий K входных символов, расположенных справа от текущей входной позиции. Название LL произошло от слова Left, поскольку анализатор просматривает входную цепочку слева направо и слова Leftmost, поскольку он обнаруживает появление правила по одному или группе символов, образующих левый край цепочки. На практике наибольшее применение имеет класс LL(1) грамматик, для которых анализатор рассматривает только один входной символ, расположенный справа от текущей позиции. Создатели ANTLR утверждают, что многие преимущества при определении действий для правил грамматики являются следствием того, что ANTLR осуществляет разбор сверху вниз. Кроме того, ANTLR выгодно отличается от других подобных программ наличием визуальной среды разработки ANTLRWorks, позволяющей удобно создавать и редактировать грамматики.

ANTLR представляет собой набор, состоящий из двух частей.

- *Генератора анализаторов* – приложения, которое получает на вход описание грамматики и генерирует код для лексического и синтаксического анализатора.
- *Библиотеки времени выполнения*, которая используется для создания конечной программы. Эта библиотека содержит базовые классы для анализаторов, а также классы, управляющие потоками символов, обрабатывающие ошибки разбора, генерирующие выходной код на основе шаблонов и многое другое.

Одна из основных задач, при решении которых используется ANTLR, - разработка трансляторов с языков DSL (Domain Specific Language – язык, специфичный для предметной области), которые генерируют код на языке высокого уровня [9].

В отличие от многих других инструментов создания компиляторов, ANTLR имеет средства для создания не только frontend-части (лексического и синтаксического анализатора), но и возможности backend (генератора кода).

При этом backend может быть организован по-разному, в зависимости от желания пользователя. В статье используется вызов действий (семантических правил, actions) в синтаксическом анализаторе. Действия вызываются синтаксическим анализатором, когда он заканчивает разбирать какую-то грамматическую конструкцию или её часть (см. рисунок 1).

На этом рисунке изображено:

- *Поток символов* – текст программы на исходном языке;
- *Lexer* – лексический анализатор, разбивающий исходный поток символов на отдельные лексем. На выходе лексический анализатор порождает поток лексем;
- *Parser* – синтаксический анализатор. Просматривает поток лексем и анализирует структуру предложений языка;

- *Actions* – действия (код). Эти действия позволяют генерировать код на языке описания Java [10] который используется в ООБД db4o;
- *Output Text* – окончательный текст на предметном языке.



Рисунок 1 - Организация элементов ANTLR

3 Представление онтологии станков и деталей машиностроительного производства в редакторе OntoStudio

На основе анализа информации, необходимой для описания станков и деталей [11-14], при построении онтологии в редакторе OntoStudio были выделены следующие классы: Станок, Деталь, Материал, Инструмент, Марка материала, Марка инструмента, Нормативные документы.

Каждый класс имеет некоторые свойства. Например, свойства класса Деталь показаны на рисунке 2. Взаимосвязи класса Деталь показаны на рисунке 3.

Local name: Деталь				
Properties				
Properties	Range	Min	Max	
Габариты детали	DT string	0	N	✗
геом.точность обрабатываемых поверхностей	DT decimal	0	N	✗
Имя детали	DT string	0	N	✗
масса (гр.)	DT integer	0	N	✗
метод изготовления	DT string	0	N	✗
Наименование детали	DT string	0	N	✗
Необходимость приспособлений	DT integer	0	N	✗
Обозначение детали	DT string	0	N	✗
Трудоемкость	DT decimal	0	N	✗
шероховатость поверхностей	DT decimal	0	N	✗
шероховатость радиусных переходов	DT decimal	0	N	✗
материал	C Материал	0	N	✗
Обрабатывается	C Станок	0	N	✗

Рисунок 2 - Свойства класса Деталь

Свойства, перечисленные на рисунке 2, могут иметь различные типы - как строка, так и целое число. Свойство «Имя детали» хранит имя детали, свойство «Наименование детали» хранит описание детали, свойство «Габариты детали» хранит габариты детали, измеренные в различных единицах измерения, поэтому было решено выбрать тип String, который в дальнейшем программным путём будет преобразовываться в нужный числовой формат.

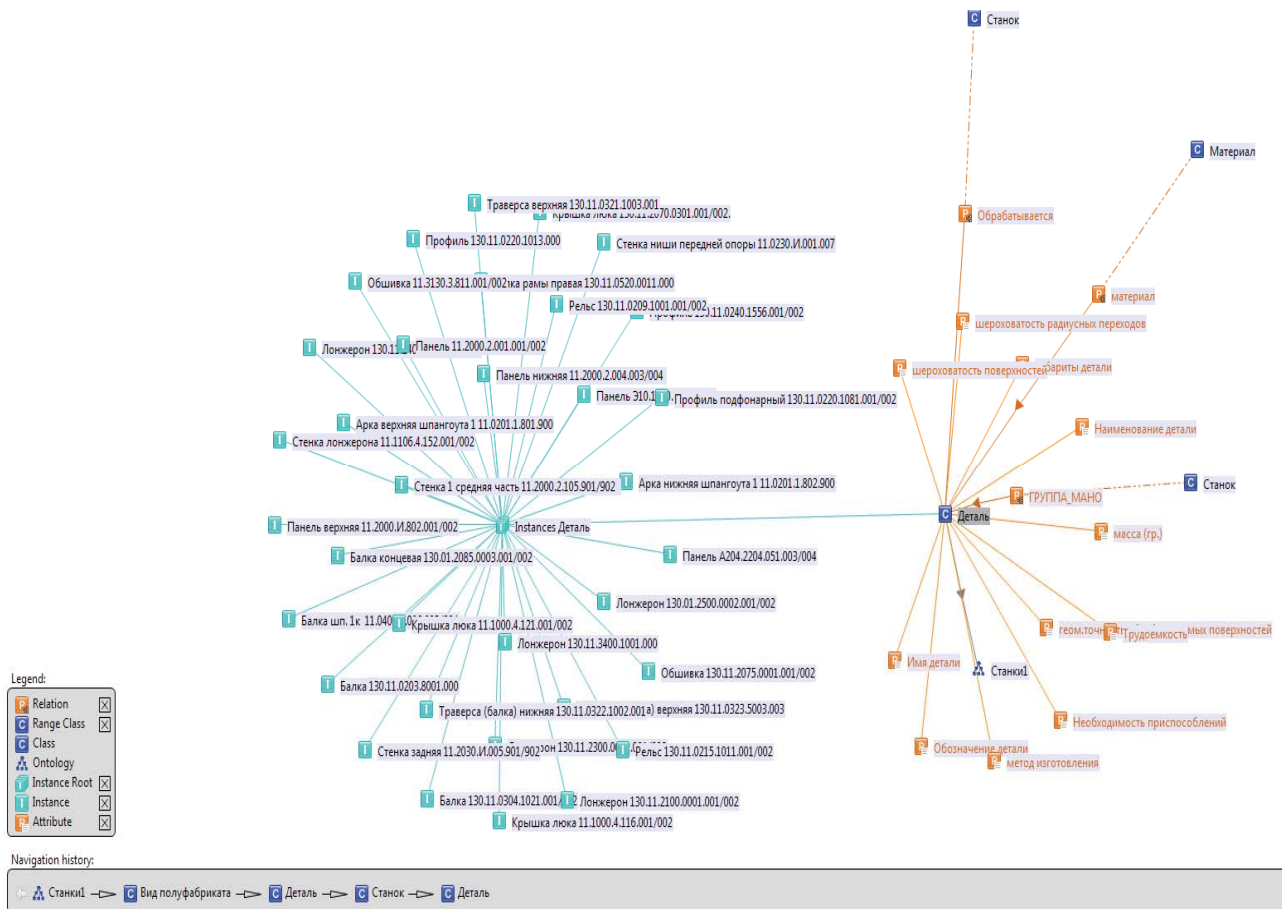


Рисунок 3 - Взаимосвязи класса Деталь

На рисунке 3 показано, как в OntoStudio можно визуальнo показатъ взаимосвязи свойств одного класса, которые являются экземплярами другого класса. В левой части рисунка изображены экземпляры класса «Деталь», а в правой части можно увидеть, какие свойства класса «Деталь» являются экземплярами других классов.

4 Структура проекта

Рассмотрим, как будет взаимодействовать ANTLR с разработками [5] и [6].
Грамматика описания онтологии на языке Object Logic приведена ниже.

```

grammar ont_db4o;
beginprog      :((name_conc ‘[(concept{ emitter.AddConc($name_conc.text );}]
schema { emitter.AddSchema($scema.text );})|
(name_conc concept_nasl{ emitter.AddConc_Child($name_conc.text);})|
name_inst(props{ emitter.AddProp($name_ist.text ); } |instance {
emitter.AddIns($name_ins.text);})|
query{ emitter.AddSQL($query );}|
rule{ emitter.AddMethod($rul.text ); } )’)+ ;
query        : ‘?’-‘ usls ;
tip : TIP|name_conc ;
usls        : ((uslo|uslp)’and’?)+;
uslo        : NP ‘.’ name_conc;
    
```

```

uslp      :      NP'[name_prop '->' (NP|CONSTC)'];
concept  '];
CONSTL   :      ('true'|'false');
name_conc
      :LET_ZAG CHAST;
concept_nasl   :      '::' name_conc ;
rule        :      uslp query;
schema     :      (name_conc '[' name_prop('{0:*}'|'1:1}')'*=>'tip']')*;
instance   :      ':'name_conc ;
name_prop  :LET_MAL (LET_MAL|LET_ZAG|INT)*;
name_inst  :name_conc INT+ ;
props     :([' name_prop '->'(CONSTC|CONSTL|TEXT) ']);
CONSTC    :      '-?'INT+('.'INT+)?;
CHAST     : (LET_MAL|LET_ZAG|INT)* ;
NP        :      '?'LET_ZAG;
TEXT      :      ""(LET_MAL|LET_ZAG|INT)+"";

```

Грамматика состоит из правил. Все правила делятся на лексические и синтаксические.

Лексические правила используются для описания лексем и служат основой для генерации лексического анализатора. Синтаксические правила описывают грамматику языка.

Лексические правила (и соответствующие им имена лексем) должны начинаться с заглавной буквы, а синтаксические – со строчной.

В грамматику вставлены Action – символы действия, позволяющие сгенерировать код на языке java.

По грамматике были сгенерированы классы:

- *Lexer* – лексический блок;
- *Parser* – синтаксический блок;
- *Emitter* – блок генерации классов db4o на языке Java.

Все эти классы экспортируются в среду разработки Eclipse [15], с помощью которой создаётся программный продукт «Конвертер Онтология – db4o». Взаимодействие всех блоков проекта Конвертер Онтология – db4o показано на рисунке 4.

Грамматика онтологии обрабатывается программой ANTLRWorks, с помощью которой генерируются классы *Lexer*, *Parser* и *Emitter*. Эти классы включаются в состав проекта приложения Конвертер Онтология – db4o на языке Java.

Файлы с онтологией в форматах OWL и RDF поступают на вход приложения. В случае наличия лексических или синтаксических ошибок выдаются сообщения. Например,

- *token recognition error at: " at 3:5*
- *mismatched input '4' expecting {'as', 'is', '[', '(', '!', ';', '+', '-', '*', '/', '%', '&', '|', '^', '<', '>', '?', '??', '++', '--', '&&', '||', '->', '==', '!=', '<=', '>=', '<<'} at 8:19*
- *extraneous input '5' expecting {'as', 'is', '[', '(', '!', ';', '+', '-', '*', '/', '%', '&', '|', '^', '<', '>', '?', '??', '++', '--', '&&', '||', '->', '==', '!=', '<=', '>=', '<<'} at 8:21*
- *no viable alternative at input 'c}' at 15:5*
- *missing '}' at 'EOF' at 15:6.*

Если ошибок нет, то генерируются классы ООБД db4o, которые можно включать в любой проект, созданный в среде разработки Eclipse.

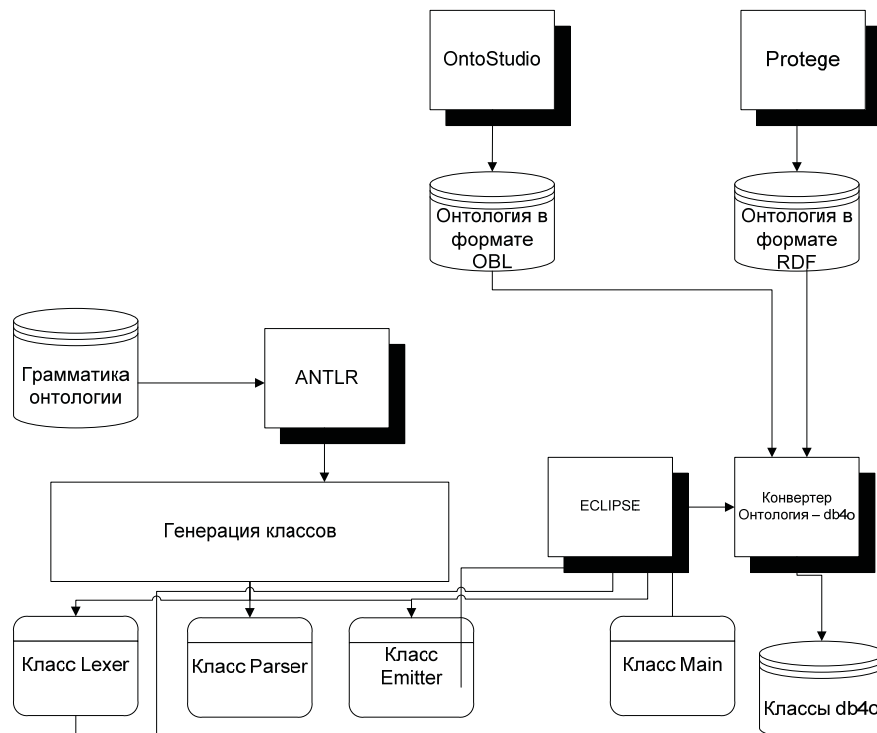


Рисунок 4 - Взаимодействие блоков проекта Конвертер Онтология – db4o

Заключение

В статье рассмотрены принципы разработки конвертера для преобразования онтологии в ООБД с использованием программного продукта ANTLR. Это позволило сократить трудозатраты при написании программного кода. Предлагаемая технология может быть распространена на случай расширения описанной онтологии, а также для обратного конвертирования из ООБД в онтологию. Для этого потребуется скорректировать грамматику ANTLR, возможно, включая класс Emitter.

Список источников

- [1] *Хорафас, Д.Н.* Системы и моделирование / Д.Н. Хорафас – пер. с англ. Е. Г. Коваленко, В. Я. Алтаева, Б. А. Квасова. - М.: Мир , 1967. - 420 с. - Пер. изд.: Systems and simulation / Dimitris N Chorafas. - New York, 1965. - 240 p
- [2] *Scott W. Ambler.* Mapping objects to relational databases. What you need to know and why. - developerWorks® IBM Corporation 2000. Отображение объектов в реляционных базах данных. - URL: <http://www.ibm.com/developerworks/library/ws-mapping-to-rdb/> (дата обращения 10.05.2016).
- [3] *Боргест, Н.М.* Онтология проектирования: теоретические основы. Часть 1. Понятия и принципы / Н.М. Боргест // Учеб. пособие. - Самара: Изд-во СГАУ, 2010. — 91 с.
- [4] *Рабчевский, Е.* Автоматическое построение онтологий. - URL: <http://shcherbak.net/avtomaticheskoe-postroenie-ontologij/> (дата обращения 02.06.2016).
- [5] *Сосинская, С.С.* Сравнение возможностей программных продуктов OntoStudio и Protege для анализа станочных систем машиностроительного производства / В.А. Игруша, С.С. Сосинская, С.А. Орсоева // Вестник ИрГТУ. - 2015. - № 3. – С. 17–23.
- [6] *Игруша, В.А.* Изоморфизм онтологии и объектной базы данных для деталей механообработки / В.А. Игруша, С.С. Сосинская// Винеровские чтения. - 2015. – С. 51–55.
- [7] *Безруков, В.* RDF — инструмент для неструктурированных данных. / В. Безруков, Ю. Конов, И. Стечкин // Открытые системы, № 9, 2012. – С.42-71.
- [8] Программный продукт ANTLR. - URL: <http://www.antlr.org/> (дата обращения 05.05.2016).

- [9] Разработка парсеров в программной среде ANTLR3. - URL: <https://theantlr.guy.atlassian.net/wiki/display/ANTLR3/FAQ+-+Getting+Started> (дата обращения 05.05.2016).
- [10] Руководство разработки Java при использовании db4o. - URL: <http://support.services.actian.com/versant/default.html> (дата обращения 10.05.2016).
- [11] **Соснин, П.И.** Структурное проектирование конфигурируемых шаблонов авиационных деталей / П.И. Соснин, О.Э. Чоракаев // Автоматизация процессов управления. - 2014. - №2 (36). - С. 99-107.
- [12] **Шустова, Д.В.** Подход к разработке семантических основ информационных систем для проектирования и производства авиационной техники / Д.В. Шустова // Онтология проектирования. - 2015. - №1 (15). - С. 70-82.
- [13] **Нитипанова, Г.П.** Онтологический анализ предметной области задачи базирования детали / Г.П. Нитипанова, С.В. Смирнов // Проблемы управления и моделирования в сложных системах: Труды XIII международной конф. (15-17 июня 2011 г., Самара, Россия). - Самара: СамНЦ РАН, 2011. - С. 85-94.
- [14] **Шведин, Б.Я.** Онтология предприятия: экспириентологический подход. Технология построения онтологической модели предприятия / Б.Я. Шведин. - М.: Ленанд. 2010. - 240 с.
- [15] Программный продукт Eclipse. - <https://eclipse.org/> (дата обращения 05.05.2016)

CONVERTER FOR THE ONTOLOGIC KNOWLEDGE BASE OF MACHINE TOOLS AND PARTS OF MACHINE-BUILDING PRODUCTION PLANT INTO AN OBJECTIVE DATABASE BUILT ON ANTLR

V.A. Igrusha¹, S.S. Sosinskaya²

Irkutsk National Research Technical University, Irkutsk, Russia

¹хоу-90@mail.ru, ²sosinskaya@mail.ru

Abstract

The article is devoted to developing an automated converter for the knowledge base in the form of ontologies represented in one of the formats, acceptable for ontology editor OntoStudio (OWL or RDF), into an object-oriented database (OODB) db4o. The proposed approach is based on the ANTLR compiler (ANother Tool for Language Recognition). This approach is implemented for the first time. It reduces labor costs and applies the ANTLR software tool to other ontologies as the developed grammar ontology for the description language is independent from the subject area. Grammar contains not only lexical and syntactical rules to verify the text, but also the rules of action for the transformation of ontology elements into elements of isomorphic object database.

Key words: *Ontology, OntoStudio, ANTLR, Protégé, converter, OODB, db4o, project, knowledge base, class.*

Citation: *Igrusha VA, Sosinskaya SS. Converter for the ontologic knowledge base of machine tools and parts of machine-building production plant into an objective database built on ANTLR [In Russian]. Ontology of designing. 2016; 3(21): 278-286. DOI: 10.18287/2223-9537-2016-6-3-278-286.*

References

- [1] *Chorafas DN.* Systems and Simulation. - New York, 1965. - 240 p.
- [2] *Scott WA.* Mapping objects to relational databases. What you need to know and why. - developerWorks® IBM Corporation 2000. <http://www.ibm.com/developerworks/library/ws-mapping-to-rdb/>.
- [3] *Borgest NM.* Ontology designing: theoretical bases. A part 1: Concepts and Principles Studies [In Russian]. - Samara: Samara State Aerospace University Publishing House, 2010. - P. 91-93.
- [4] *Rabchevsky E.* Automatic construction of ontologies [In Russian]. Source: <http://shcherbak.net/avtomaticheskoe-postroenie-ontologij/>.
- [5] *Sosinskaya SS, Igrusha VA, Orsoeva SA.* Comparison of possibilities of software products OntoStudio and Protege for the analysis of machine systems of machine-building manufacture [In Russian]. The Bulletin of Irkutsk National Research Technical University. - 2015; 3:17-23.

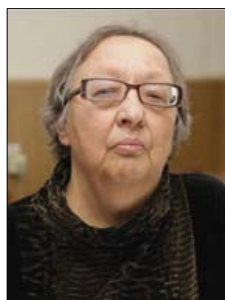
- [6] *Igrusha VA, Sosinskaya SS*. Isomorphism of ontology and objective database for details mechanical engineering [In Russian]. *Vinerovsky readings*. - 2015. - P. 51-55.
 - [7] *Bezrukov V, Konov Yu, Stechkin I*. RDF – the tool for not structured data [In Russian]. *Open systems*. - 2012; 9: 42-71.
 - [8] The software ANTLR. Source: <http://www.antlr.org/>.
 - [9] The development of parsers in ANTLR3 software environment. Source: <http://www.antlr3.org/>.
 - [10] Guide Java development using db40. Source: <http://supportservices.actian.com/versant/default.html/>.
 - [11] *Sosnin PI, Chorakaev OE*. Structural designing of configured templates of aviation parts [In Russian]. *Automation of management processes*. – 2014; 2(36): 99-107.
 - [12] *Shustova DV*. The approach to the working out of semantic bases of information systems for the designing and manufacture of aviation technics [In Russian]. *Ontology of designing*. – 2015; 1(15): 70-82.
 - [13] *Nitipanova GP, Smirnov SV*. The ontologic analysis of a subject domain of a problem of basing of a detail [In Russian]. *Problems of management and modeling in difficult systems: Work of XIII International Conf. (15 June 17, 2011, Samara, Russia)*. - Samara: SamNTs RAS, 2011. - P. 85-94.
 - [14] *Shvedin BYa*. The ontology the enterprise: experientological approach. *Technology of construction of ontological model of the enterprise* [In Russian]. – М.: LENAND. 2010. - P. 240.
 - [15] Eclipse. Source: <https://eclipse.org/>.
-

Сведения об авторах



Игруша Владимир Андреевич, 1990 г. рождения. Окончил Иркутский национальный исследовательский технический университет. (2013). Аспирант кафедры технологии и оборудования машиностроительных производств. Область научных интересов – онтологии, базы данных. Имеет около 10 публикаций. Инженер-программист 2 категории на Иркутском авиационном заводе - филиал ПАО "Корпорация "Иркут".

Igrusha Vladimir Andreevich, (b 1990) He graduated from Irkutsk national research technical University. (2013). Post-graduate student of department of technology and the equipment of machine-building manufactures. Research interests – ontology, database. He has about 10 publications. The Engineer-programmer of 2 categories on Irkutsk Aviation Factory - branch PAO "Corporation "Irkut".



Сосинская Софья Семеновна, окончила Иркутский государственный университет по специальности «Математик-вычислитель». Кандидат технических наук, профессор кафедры технологии и оборудования машиностроительных производств. Область научных интересов – искусственный интеллект, трансляторы. Имеет свыше 70 публикаций.

Sosinskaya Sofya Semenovna, graduated from Irkutsk state university with the "Mathematician-calculator" speciality. PhD in Technical Science, Professor of department of technology and the equipment of machine-building manufactures. Research interests - artificial intelligence, translators. She has over 70 publications.