

УДК 004.048

ФОРМАЛИЗАЦИЯ ОПИСАНИЯ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ ИЗГОТОВЛЕНИЯ ДЕТАЛЕЙ МАШИНОСТРОЕНИЯ НА ОСНОВЕ ОНТОЛОГИИ И ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ БАЗ ДАННЫХ

В.А. Игруша¹, С.С. Сосинская²

Иркутский национальный исследовательский технический университет, Иркутск, Россия

¹хоу-90@mail.ru, ²sosinskaya@mail.ru

Аннотация

Статья посвящена разработке структурного представления о технологических процессах изготовления деталей машиностроения на основе онтологии и объектно-ориентированных баз данных. На основе общности объектно-фреймового стиля представления знаний для формальных онтологий и для объектно-ориентированных баз знаний, рассмотрен автоматизированный перевод с языка онтологий в объектно-ориентированные базы данных с помощью программного средства ANTLR. Построена онтология технологических процессов в машиностроении на основе анализа диаграммы бизнес-процессов разработки технологии и её декомпозиции. Онтология представлена на языке RIF, разновидности XML. Создана грамматика языка RIF и класс EMITTER для генерации кода на языке объектно-ориентированной базы данных db4o. Такая методика позволяет, описывая технологический процесс изготовления деталей на языке онтологий, исполнять запросы как методы классов db4o, что даёт сокращение временных затрат и повышает эффективность такой комбинированной системы по сравнению с отдельным использованием редактора OntoStudio или db4o.

Ключевые слова: онтология, OntoStudio, ANTLR, конвертер, бизнес-процесс, база знаний, класс.

Цитирование: Игруша, В.А. Формализация технологических процессов изготовления деталей машиностроения на основе онтологии и объектно-ориентированных баз данных/ В.А. Игруша, С.С. Сосинская // Онтология проектирования. – 2017. – Т. 7, №1(23). – С. 77-88. – DOI: 10.18287/2223-9537-2017-7-1-77-88.

Введение

В области машиностроения разработано много информационных моделей представления данных в виде различных структур, однако необходимо искать новые подходы и выбирать такие структуры, которые позволяют динамически пополнять множество запросов, которые возникают постоянно по мере эксплуатации модели. Для решения этой проблемы авторами был предложен новый подход, сочетающий формирование онтологии, содержащей информацию о технологических возможностях станков и обрабатываемых на них деталей [1], и одновременно с этим реализацию возможности хранения той же информации в виде объектно-ориентированной базы данных (ООБД) [2]. В данной работе предлагается распространить этот подход на более широкую предметную область (ПрО) – разработка технологии изготовления деталей.

Онтология – один из способов структурирования знаний путём создания иерархии классов с их свойствами, правилами и запросами. Классы вместе с их экземплярами создают объектную базу знаний [3]. Правила позволяют выводить новые факты [4]. С другой стороны, ООБД также представляют собой набор классов, образующих иерархию и содержащих свойства, методы, запросы и объекты (экземпляры) классов [5].

В статье предложен итерационный процесс построения иерархии объектов на основе методики анализа бизнес-процессов задачи [6]. Разработка структур представления базы знаний позволила сравнить выполняемые на их основе запросы по скорости обработки, а также сделала возможным переход от одного представления к другому для выбора наиболее эффективного для различных пользователей.

1 Методика разработки технологических процессов изготовления деталей на предприятиях

Технологический процесс (ТП) [7] разрабатывается для изготовления новой детали или совершенствования выпускаемой. Основой для нового ТП обычно служит имеющийся типовой или групповой ТП. Если таковые отсутствуют, то за основу берут действующие единичные ТП изготовления аналогичных деталей.

Разработка ТП начинается с анализа исходных данных. Необходимо по имеющимся сведениям о программе выпуска и конструкторской документации на новую деталь ознакомиться с её назначением и конструкцией, требованиями к изготовлению и эксплуатации.

Затем последовательно выбирают действующий типовой, групповой ТП или аналог единичного процесса, наиболее близкий к разрабатываемому ТП в соответствии с технологическим классификатором.

По классификатору заготовок, методике расчёта и технико-экономической оценке ТП выбирают исходную заготовку и методы её изготовления, даётся технико-экономическое обоснование выбора заготовки.

Затем проектируют технологические операции и рассчитывают режимы обработки, составляют последовательность переходов в каждой операции, выбирают средства технологического оснащения (СТО), в том числе средства контроля и испытаний, используя стандарты, каталоги, альбомы.

На этом же этапе выбирают средства механизации и автоматизации процесса и внутрицеховые средства транспортирования, назначают и рассчитывают режимы обработки на основании технологических нормативов.

Затем осуществляется нормирование ТП: устанавливаются исходные данные для расчёта норм времени и расхода материалов, рассчитываются затраты труда и расход материалов, исходя из разряда работ и профессии исполнителей операций с использованием соответствующих нормативов и классификаторов.

После этого выбирают из нескольких возможных вариантов тот ТП, который является наиболее дешёвым с точки зрения используемых материалов и наименее трудозатратным.

На заключительном этапе, на основании стандартов Единой системы технологической документации (ЕСТД), оформляется документально ТП, осуществляется нормоконтроль технической документации.

2 Актуальность хранения сложной информации о технологиях в виде единой структуры

Обычно, на предприятиях используется несколько информационных систем, в рамках которых разрабатываются технологии изготовления деталей. В качестве примера можно привести систему ЛОЦМАН:PLM [8], предназначенную для управления инженерными данными и жизненным циклом изделия, в которую включён полный цикл технологии изготовления изделия в виде единой структуры.

По мере наполнения состава изделия конструкторскими данными технологи, используя САПР ТП ВЕРТИКАЛЬ [9], начинают технологическую проработку конструкции, определяют маршрут изготовления в системе ЛОЦМАН:PLM *Расцеховщик*, согласуют конструкторскую документацию и оценивают потребность в СТО. По окончании работы технологических служб предприятия в системе ЛОЦМАН:PLM формируются следующие данные:

- разработанный и утверждённый ТП;
- нормы расхода основных и вспомогательных материалов;
- нормы времени на изготовление деталей и сборку узлов;
- спроектированные чертежи вспомогательной оснастки и инструмента.

3 Этапы разработки объектной модели

В модели ТП, соответствующей методике разработки ТП на предприятии, будут указываться различные конструкторско-технологические документы, создаваться извещения об изменении в ТП, включая в него операции. Для выполнения операций технологи должны выбирать:

- станки, на которых будет обрабатываться деталь;
- технологические карты неразрушающего контроля;
- материалы, которые будут использоваться при изготовлении;
- инструментальные наладки;
- технологические эскизы изделий;
- технологические переходы (в переходах необходимо выбирать вид оснастки).

На основе этих предположений была разработана диаграмма бизнес-процесса верхнего уровня на основе стандарта IDEF0 [10], показанная на рисунке 1.

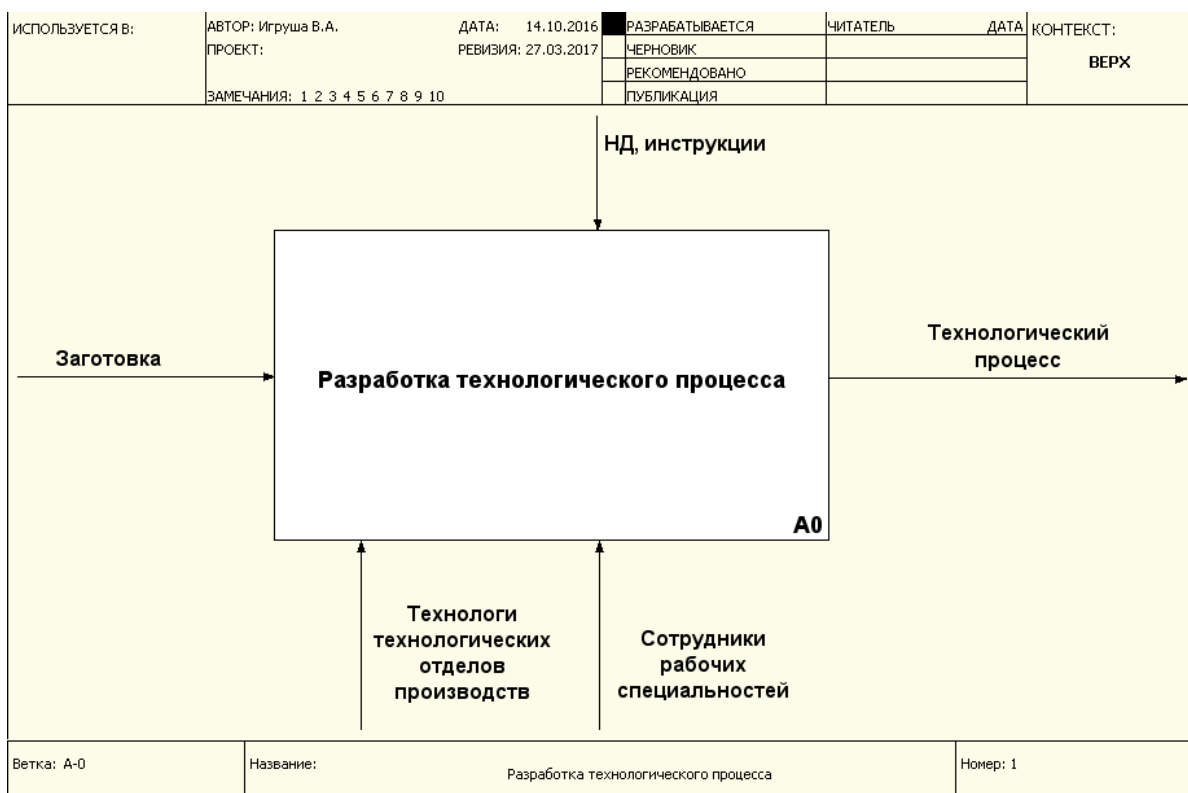


Рисунок 1 - Диаграмма верхнего уровня процесса «Разработка ТП изготовления детали»

Новизна подхода, предлагаемого авторами, состоит в том, что входы и выходы этой диаграммы могут рассматриваться в качестве классов ПрО. На основании этой диаграммы, наполняя классы атрибутами, получаем UML-диаграмму [11] классов «Деталь-ТП», изображённую на рисунке 2. Диаграмма декомпозиции бизнес-процесса верхнего уровня показана на рисунке 3.

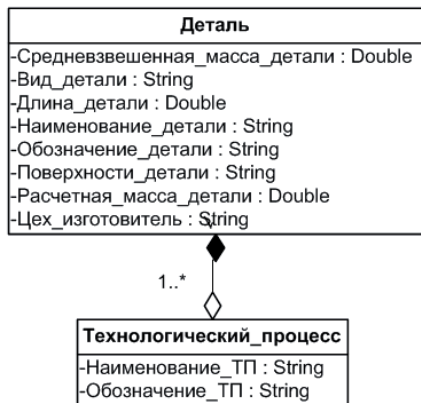


Рисунок 2 - Начальная диаграмма классов «Деталь - ТП»

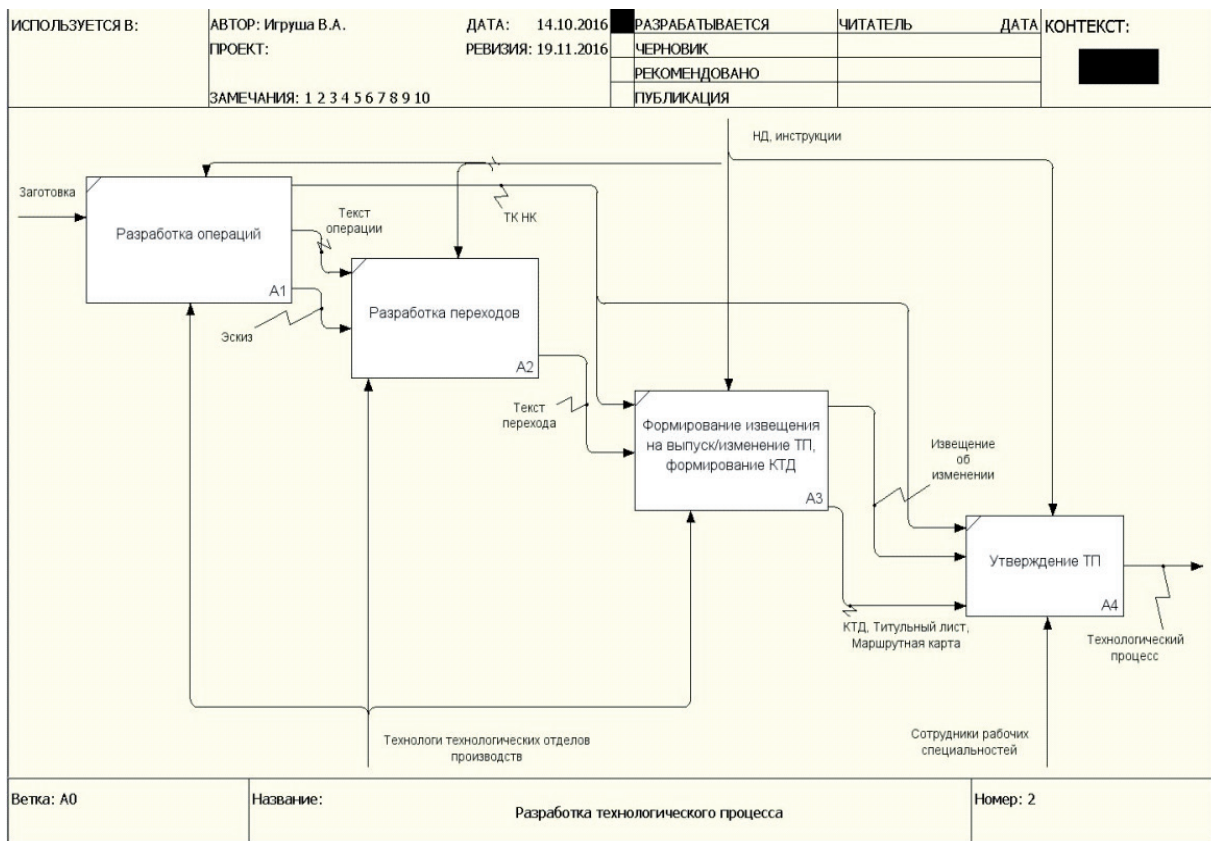


Рисунок 3 - Диаграмма декомпозиции процесса «Разработка ТП изготовления детали»

В результате анализа входов и выходов этой диаграммы и наполнения классов атрибутами, получаем диаграмму классов «Деталь - ТП»; она изображена на рисунке 4.

Полученная диаграмма классов может рассматриваться в качестве модели ПрО как для построения онтологии, так и для построения ООБД.

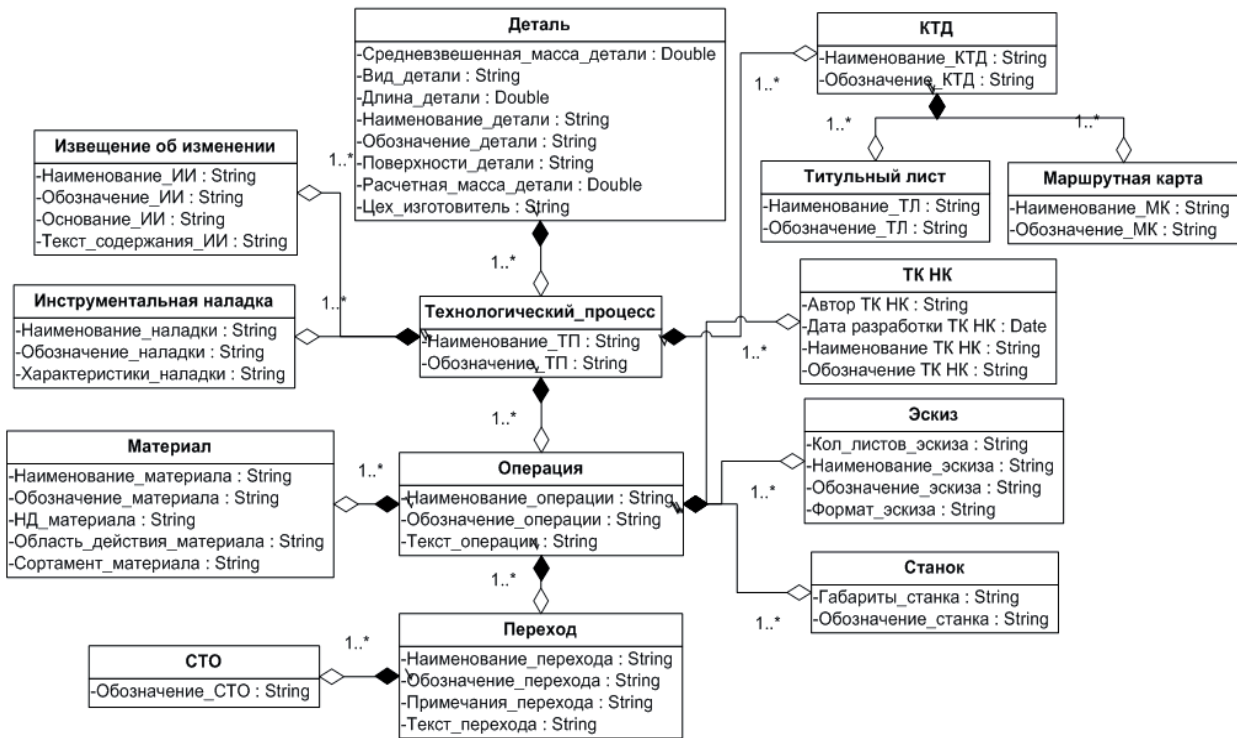


Рисунок 4 - Уточненная диаграмма классов «Деталь – ТП»

4 Обоснование выбора онтологии и ООБД для решения задачи

Одним из преимуществ использования онтологий в качестве инструмента организации знаний является системный подход к моделированию ПрО. При этом достигаются:

- систематичность — онтология представляет целостный взгляд на ПрО;
- единообразие — информация, представленная в единой форме, гораздо лучше воспринимается;
- логическая непротиворечивость — построение онтологии позволяет восстановить недостающие логические связи во всей их полноте.

При разработке онтологий основным является создание классов, образующих иерархию, и наполнение их свойствами, которые могут быть как простыми полями, так и экземплярами других классов.

ООБД – базы данных, в которых информация представлена в виде объектов, как в объектно-ориентированных языках программирования, т.е. содержат свойства и методы. Для манипулирования объектами разработаны системы управления ООБД (СУООБД), которые могут хранить произвольное количество различных объектов, связанных отношениями иерархического наследования. Использование реляционной модели приводит к созданию многочисленных таблиц¹, при этом при выполнении запросов приходится постоянно организовывать объединения таблиц. Классы являются моделью более высокого уровня по сравнению с реляционными таблицами для отображения объектов реального мира.

В качестве программного средства описания рассматриваемой задачи был выбран программный продукт *OntoStudio* [12] - среда моделирования для создания и поддержания

¹ Это верно при использовании классического подхода к проектированию реляционных баз данных (РБД), когда приоритет отдаётся критериям, далёким от задач семантического моделирования ПрО (это экономия памяти приводит к хранению фрагментов информационного образа объекта ПрО в различных таблицах). «Семантизация» проектирования РБД снимает проблемы реляционной модели (по крайней мере, те, что указывают авторы) – *Прим. ред.*

онтологий. В качестве СУООБД нами была выбрана db4o [13]. Эта система является встраиваемой в различные языки, она поставляется в виде свободно распространяемых библиотек (имеются реализации для Java, C# и других языков). Кроме этого имеется программный редактор ObjectManager 7.4 [14], с помощью которого можно получить представление о модели ООБД (классы, их атрибуты и их типы). Можно также выполнять запросы на языке SQL [15].

db4o поддерживает три запросные системы:

- Query-By-Example (QBE);
- Native Queries (NQ);
- Simple Object Database Access (SODA) API.

При использовании QBE необходимо создать шаблон. db4o выберет все объекты, которые соответствуют всем значениям полей, не имеющих значений по умолчанию. Это делается путём отображения всех полей и построения запросного выражения, в котором все поля, не имеющие значения по умолчанию, комбинируются с помощью выражений AND.

Пример:

```
//Выбор деталей по имени
Detail washer_det= new Detail("Шайба", 0);
ObjectSet result = db.QueryByExample(washer_det);
ListResult(result);
```

Запросы Native – это главный запросный интерфейс db4o. Так как запросы Native просто используют семантику языка программирования, они стандартизованы и безопасны.

Пример:

```
//Выборка станков по имени детали.
string detailName = " Шайба";
ObjectSet results = db.Query(new RetrieveMachineByDetailNamePredicate(detailName));
ListResult(results);
```

Запросы SODA являются нижним уровнем запросов db4o, позволяющим осуществить прямой доступ к узлам графа запросов. Так как SODA использует строки для идентификации полей, эти запросы являются довольно громоздкими.

Пример:

```
//Выборка всех деталей.
IQuery query = db.Query();
query.Constrain(typeof(Detail));
ObjectSet result = query.Execute();
ListResult(result);
```

5 Запросы и правила в OntoStudio

В программном редакторе OntoStudio есть модули создания запросов и правил. Запросы играют важную роль для отбора нужных данных, каждый запрос выделяет подмножество объектов с определёнными условиями.

Приведём несколько примеров применительно к рассматриваемой ПрО.

Запрос 1: найти детали, которые обрабатываются с переходом "Выдержать деталь в ванне для промывки" в ТП.

Запрос на языке Objectlogic:

```
@{options[outorder(?Наименование_детали,?Обозначение_ТП,?Обозначение_перехода,?
Обозначение_операции)]}
?-
```

?Переход:Переход[Обозначение_перехода->?Обозначение_перехода] and
 ?Технологическая_операция:Операция[Обозначение_операции->?Обозначение_операции,Переход_операции->?Переход] and
 ?Технологический_процесс:Технологический_процесс[Обозначение_ТП->?Обозначение_ТП,Операция_ТП->?Технологическая_операция] and
 ?Деталь:Деталь[Наименование_детали->?Наименование_детали,ТП_детали->?Технологический_процесс] and
 _contains(?Обозначение_перехода,"Выдержать деталь в ванне промывки").

На рисунке 5 показан результат выполнения Запроса 1.

?Наименование_детали	?Обозначение_ТП	?Обозначение_перехода	?Обозначение_операции
"Траверса верхняя"	"208.01102.00005"	"Выдержать деталь в ванне промывки"	"208_001933_А"
"Балка"	"208.01102.00004"	"Выдержать деталь в ванне промывки"	"208_001933_А"
"Балка"	"208.01102.00001"	"Выдержать деталь в ванне промывки"	"208_001933_А"
"Рельс"	"208.01102.00001"	"Выдержать деталь в ванне промывки"	"208_001933_А"

Рисунок 5 – Результат выполнения запроса 1

Рассмотрим взаимодействие правил вывода и запросов в редакторе OntoStudio.

Правило вывода формализует процесс рассуждения, в ходе которого осуществляется переход от некоторых исходных фактов (посылок) к новым фактам — заключениям. Таким образом, наличие правил вывода расширяет базу знаний новыми фактами.

Правило 1: если для детали определена операция «отрубка», «надрубка», «разрубка», то она имеет вид «Полуфабрикат».

Правило1 на языке Objectlogic:

?Деталь[Вид_детали->"Полуфабрикат"] :- ?Деталь:Деталь[Операция_детали->"отрубка" or Операция_детали->"надрубка" or Операция_детали->"разрубка"].

При наличии такого правила выполнение *запроса 2:* поиск деталей, имеющих наименование «Балка», вид «Полуфабрикат» и пять обрабатываемых поверхностей, даст непустой результат. Изначально в базе знаний отсутствовали экземпляры деталей вида «Полуфабрикат».

Запрос 2 на языке Objectlogic:

@{options[outorder(?Обозначение_детали,?Наименование_детали,?Вид_детали,?Поверхности_детали)]}
 ?-?Экземпляр_Детали:Деталь[Обозначение_детали->?Обозначение_детали,Поверхности_детали->?Поверхности_детали,Вид_детали->?Вид_детали] and _unify(?Наименование_детали,"Балка") and _unify(?Вид_детали,"Полуфабрикат") and _unify(?Поверхности_детали,"5").

Рассмотрим еще один пример с применением правил 2 и 3 и запроса 3.

Правила 2 и 3 определяют «Цех_изготовитель» детали по её длине:

если длина детали =10, то цех=245,

если длина =7, то цех=244.

Правило 2 на языке Objectlogic:

?Деталь[Цех_изготовитель->"245"] :- ?Деталь:Деталь[Длина_детали->"10"].

Запрос 3: Необходимо найти все детали, у которых «Цех_изготовитель» = 245 или 244, «Наименование_детали» = «Балка» и «Поверхности_детали» = 5. В этом запросе участвуют значения свойства «Цех_изготовитель», сформированные на основе правил 2 и 3.

Запрос на языке Objectlogic:

@{options[outorder(?Обозначение_детали,?Наименование_детали,

```
?Цех_изготовитель,?Поверхности_детали,?Экземпляр_Детали)]}
?-?Экземпляр_Детали:
Деталь[Обозначение_детали->?Обозначение_детали,
Поверхности_детали->?Поверхности_детали,
Цех_изготовитель->?Цех_изготовитель]and
unify(?Наименование_детали,"Балка")
and (_unify(?Цех_изготовитель,"245") or _unify(?Цех_изготовитель,"244")) and
_unify(?Поверхности_детали,"5").
```

6 Использование инструмента ANTLR для конвертирования онтологии в ООБД

Аналогично [3] будем использовать методику конвертирования онтологии в базу данных ООБД db4o на основе инструментария для создания компиляторов ANTLR [16]. Этот инструмент способен не только осуществлять лексический и синтаксический анализ обрабатываемого текста, но и генерацию кода программы на языке Java (или на другом из predefined языков). В отличие от [3] будем сохранять онтологию в формате RIF [17]. Это формат XML, в котором различные теги соответствуют основным объектам онтологии. Так, тег <sentence> начинает каждое предложение; тег <Frame> описывает классы и экземпляры классов, тег <slot> описывает свойства, тег <forall> описывает запросы и т.д. Такой формат более удобен для включения в грамматику входного языка (в данном случае RIF) семантических атрибутов (attributes) и семантических действий (actions), позволяющих сгенерировать код на языке Java. Методы генерации кода включаются в специальный класс, который для определенности назван Emitter.

Классы, сгенерированные с помощью ANTLR, и класс Emitter включаются в состав проекта NetBeans и позволяют сгенерировать классы ООБД db4o.

Часть разработанной грамматики описания онтологии рассматриваемой ПрО на языке RIF приведена ниже:

```
grammar ont_db4o_rif;
prog      :(SENTENCE sentence ESENTENCE)+ ;
sentence
    :Frame OBJECT const_class EOBJECT EFRAME
    |ATOM OP const_card EOP ARG const_class const_prop CONST_MIN CONST_MAX EARG EATOM
    |ATOM OP const_range EOP ARG const_class const_prop const_type EARG EATOM;
const_class :CONST_CLASS letdigsymb name_class=ident_class
{ emitter.Add_Class($name_class.text); } ECONST;
ident_class : letter_zag chast;
ident_prop  : letter chast;
const_card  : CONST_CARD ECONST ;
CONST_CLASS:'<Const type="http://www.w3.org/2007/rif#iri">http://www.';
CONST_MIN  :'<Const type="&xsd:int">1</Const>';
CONST_CARD:'<Const
type="http://www.w3.org/2007/rif#iri">http://www.ontoprise.com/2009/03/rif/obld#propertyCardinality';
CONST_MAX  :('<Const type="&xsd:int">1</Const>|
<Const type="&xsd:int">-1</Const>');
const_type : '<Const type="http://www.w3.org/2007/rif#iri">&xsd;' name_type=type
{ emitter.Add_Type($name_type.text); } ECONST;
fragment NUMB :'#';
CONST_RANGE:'<Const
type="http://www.w3.org/2007/rif#iri">http://www.ontoprise.com/2009/03/rif/obld#propertyRange<';
const_prop:CONST_CLASS letdigsymb NUMB name_prop=ident_prop
{ emitter.Add_Prop($name_prop.text); } ECONST;
```



```

letdigisymb:'NewOnto1.org/ontology1';
letter:      letter_zag|letter_small;
letter_small:'a'..'z';
letter_zag:'A'..'Z';
fragment INT : '0'..'9';
type        :TYPE|ident_class;
TYPE
  :'_int'
  |'_integer'
  |'_decimal'
  |'_double'
  |'_string'
  |'_date'
  |'_time'
  |'_dateTime'
  |'_boolean'      ;
chast       :(letter_small|letter_zag|INT)*;
fragment SENTENCE:'<sentence>';
fragment ESENTENCE : '</sentence>';
fragment FRAME   : '<Frame>';
fragment EFRAME  : '</Frame>';
fragment OBJECT  : '<object>';
fragment EOBJECT : '</object>';
fragment ATOM    : '<Atom>';
fragment EATOM   : '</Atom>';
fragment ECONST  : '</Const>';
fragment OP      : '<op>';
fragment EOP     : '</op>';
fragment ARG     : '<args ordered="yes">';
fragment EARG    : '</args>';

```

7 Сравнение временных затрат при выполнении запроса в OntoStudio и в ООБД db4o

Сформируем простой запрос: найти все детали, имеющиеся в базе знаний. Этот запрос в программном редакторе онтологий OntoStudio имеет вид:

```

?-?Экземпляр_Детали:Деталь[Обозначение_детали->?Обозначение_детали,
Поверхности_детали->?Поверхности_детали,Вид_детали->?Вид_детали].

```

Временные затраты при выполнении этого запроса показаны на рисунке 6.

Результат запроса в db4o с помощью запросной системы SODA показан на рисунке 7. Переведа время в обоих случаях в миллисекунды, получим: db4o - 1964 ms, OntoStudio - 2268 ms. Таким образом, запрос в db4o выполнен быстрее, чем в OntoStudio в 1.154 раза.

Сравнение позволяет сделать предварительный вывод, что предложенная технология разработки онтологии с её последующим конвертированием в ООБД более эффективна по времени выполнения запроса.

Для подтверждения эффективности предложенной технологии требуется провести исследование для широкого спектра запросов.

Заключение

В статье рассмотрены этапы построения онтологии для описания технологий изготовления деталей в их связи с диаграммой бизнес - процессов, целесообразность совместного использования онтологий и объектно-ориентированных баз данных,

эффективность применения программного продукта ANTLR для автоматизированного преобразования из одного представления в другое. Показано более эффективное по времени выполнение запроса в ООБД db4o по сравнению с редактором онтологий OntoStudio.

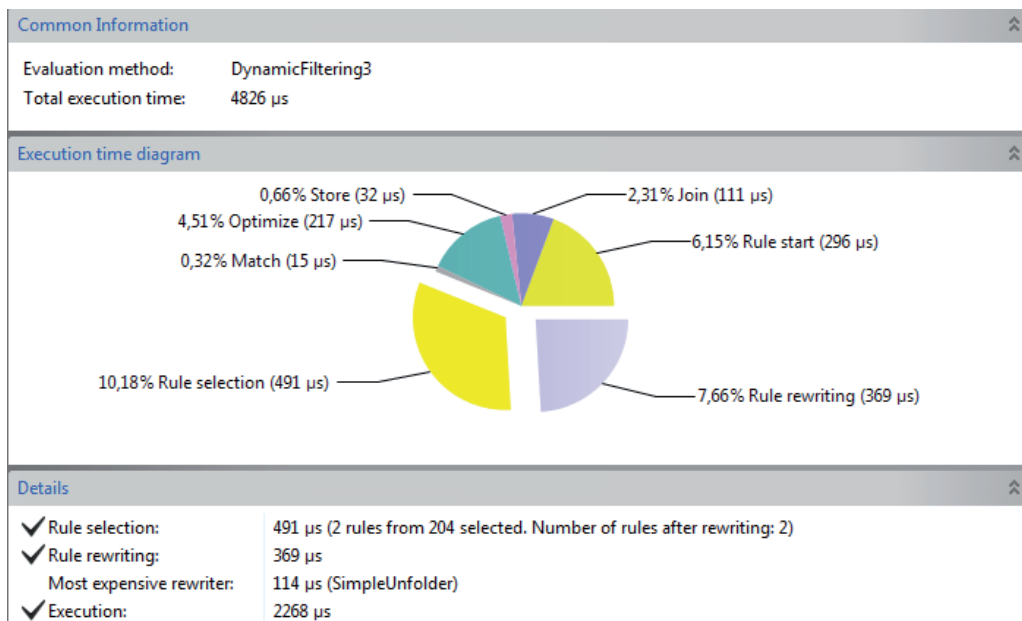


Рисунок 6 – Результат выполнения запроса в OntoStudio

```
//Выборка всех Деталей.
IQuery query = db.Query();
query.Constrain(typeof(Detail));
IObjectSet result = query.Execute();
ListResult(result);
```

Problems Javadoc Declaration Debug Console

work (1) [Java Application] C:\Program Files (x86)\Java\jre6\bin\javaw.exe (04.10.2016 16:25:28)

Время выполнения программы: 1964432НаноСек

Рисунок 7 – Запрос и результат его выполнения в db4o

Список источников

- [1] **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.
- [2] **Игруша, В.А.** Изоморфизм онтологии и объектной базы данных для деталей механообработки / В.А. Игруша, С.С. Сосинская // Винеровские чтения. - 2015. – С. 51–55.
- [3] **Игруша, В.А.** Конвертер онтологической базы знаний станков и деталей машиностроительного производства в объектную базу данных на основе ANTLR / 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.
- [4] **Тяпкин, С.В.** Преимущества и недостатки объектно-ориентированных баз данных / С.В. Тяпкин // Журнал научных публикаций аспирантов и докторантов. 2007. – URL: <http://jurnal.org/articles/2007/inf3.html> (дата обращения 21.12.2016)
- [5] **Кузнецов, С.** Объектно-ориентированные базы данных - основные концепции, организация и управление: краткий обзор / С. Кузнецов// - URL http://citforum.ru/database/articles/art_24.shtml#lit (дата обращения 21.12.2016).

- [6] **Репин, В.** Методики анализа бизнес-процессов. Глава из книги «Процессный подход к управлению / В. Репин, В. Елиферов // Моделирование бизнес-процессов». М.: Издательство "Манн, Иванов и Фербер".
- [7] **Вагнер, В.А.** Технологические процессы в машиностроении / В.А. Вагнер // Барнаул: Изд-во АлтГТУ, 2006. - 592 с.
- [8] ЛОЦМАН:PLM. – URL: <http://machinery.ascon.ru/software/developers/items/?prpid=889> (дата обращения 22.12.2016).
- [9] Система автоматизированного проектирования технологических процессов ВЕРТИКАЛЬ. – URL: <http://machinery.ascon.ru/software/tasks/items/?prcid=8&prpid=420> (дата обращения 22.12.2016)
- [10] **Верников Г.** Корпоративный менеджмент / Г. Верников // – URL: <http://www.cfin.ru/vernikov/idef/idef0.shtml> (дата обращения 22.12.2016).
- [11] **Новиков, Ф.А.** Моделирование на UML. Теория, практика, видеокурс / Ф.А. Новиков, Д.Ю. Иванов // – СПб.: Профессиональная литература, Наука и Техника, 2010.
- [12] OntoStudio. – URL: <http://www.semafora-systems.com/en/products/ontostudio/> (дата обращения 22.12.2016).
- [13] Руководство разработки Java при использовании db4o. – URL: <https://www.dropbox.com/s/s6farl3dqe7e1r/db4o-8.0.276.16149-java.zip?dl=0> (дата обращения 10.05.2016).
- [14] ObjectManager. – URL: <http://www.cnmosoft.com/Мac/db4o-object-manager-251021.html> (дата обращения 22.12.2016).
- [15] SQL. – URL: <http://www.sql.ru/> (дата обращения 22.12.2016).
- [16] Программный продукт ANTLR. - URL: <http://www.antlr.org/> (дата обращения 05.05.2016).
- [17] RIF Overview (Second Edition) W3C Working Group Note 5 February 2013. – URL: <http://www.w3.org/TR/rif-ld/> (дата обращения 22.12.2016).

FORMALIZATION OF DESCRIPTION OF TECHNOLOGICAL PROCESSES FOR MANUFACTURING OF MACHINE-BUILDING PARTS ON THE BASIS OF ONTOLOGY AND OBJECT-ORIENTED DATABASES

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 the development of a structural representation of the technological processes of manufacturing machine-building parts on the basis of ontology and object-oriented databases. Based on the generality of the object-frame representation of knowledge for formal ontologies, and for object-oriented knowledge bases, an automated translation from the ontology language into object-oriented databases is considered using the ANTLR software tool. The ontology of technological processes in machine building is constructed on the basis of the analysis of the business process diagram of technology development and its decomposition. The ontology is represented in the RIF language, a kind of XML. A grammar of the RIF language and the EMITTER class have been created to generate code in the language of the object-oriented database db4o. Such a technique allows, when describing the technological process of manufacturing parts in the ontology language, to execute queries as methods of classes db4o, which reduces the time costs and improves the efficiency of such a combined system in comparison with the separate use of the editor OntoStudio or db4o..

Key words: ontology, OntoStudio, ANTLR, converter, db4o, business process, knowledge base, class.

Citing: Igrusha VA, Sosinskaya SS. Formalization of description of technological processes for manufacturing of machine-building parts on the basis of ontology and object-oriented databases. *Ontology of designing*. 2017; 7(1): 77-88. DOI: 10.18287/2223-9537-2017-7-1-77-88.

References

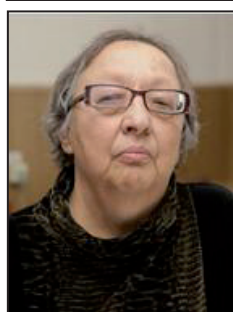
- [1] *Sosinskaya SS, Igrusha VA, Orsoeva SA*. Comparison of possibilities of software products OntoStudio and Protege for the analysis of machine systems of machine-building manufacture [In Russian]. The Bulletin of Irkutsk National Research Technical University. – 2015. - №3:17-23.
- [2] *Igrusha VA, Sosinskaya SS*. Isomorphism of otology and objective database for details mechanical engineering [In Russian]. Vinerovsky readings. - 2015. - P. 51-55.
- [3] *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; 6(3): 278-286. DOI: 10.18287/2223-9537-2016-6-3-278-286.
- [4] *Тяпкин SV*. Advantages and disadvantages of object-oriented databases [In Russian]. Journal of scientific publications of postgraduate students and doctoral students. 2007. - <http://jurnal.org/articles/2007/inf3.html>.
- [5] *Kuznetsov S*. Object-oriented databases-basic concepts, organization and management: a brief review [In Russian]. - http://citforum.ru/database/articles/art_24.shtml#lit.
- [6] *Репин V., Елиферов V*. Methods of analyzing business processes [In Russian]. Chapter from the book "Process approach to management. Modeling of business processes". M.: Publishing house "Mann, Ivanov and Ferber", 2013.
- [7] *Wagner VA*. Technological processes in mechanical engineering [In Russian]. Barnaul: Publishing house of AltSTU, 2006. - 592 P.
- [8] LOTSMAN: PLM. - <http://machinery.ascon.ru/software/developers/items/?prpid=889>.
- [9] Computer-aided design of technological processes VERTICAL. - <http://machinery.ascon.ru/software/tasks/items/?prcid=8&prpid=420>.
- [10] *Vernikov G*. Corporate Management [In Russian]. - <http://www.cfin.ru/vernikov/idef/idef0.shtml>.
- [11] *Novikov FA, Ivanov DYu*. Modeling in UML. Theory, practice, video course [In Russian]. - SPb.: Professional Literature, Science and Technology, 2010.
- [12] OntoStudio. - <http://www.semafora-systems.com/en/products/ontostudio/>.
- [13] Guide Java development using db40. - <https://www.dropbox.com/s/s6farl3dqe7e1r/db4o-8.0.276.16149-java.zip?dl=0>.
- [14] ObjectManager. - <http://www.cnmosoft.com/Mac/db4o-object-manager-251021.html>.
- [15] SQL. - <http://www.sql.ru/>.
- [16] ANTLR. - <http://www.antlr.org/>.
- [17] RIF Overview (Second Edition) W3C Working Group Note 5 February 2013. - <http://www.w3.org/TR/rif-fld/>.

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 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.