

О ВОЗМОЖНОСТИ ОНТОЛОГИЧЕСКОГО ОПИСАНИЯ ТЕХНИЧЕСКИХ ТРЕБОВАНИЙ К ЭТАПАМ ЖИЗНЕННОГО ЦИКЛА АВТОМОБИЛЯ «КАМАЗ»

Ипатов Д.А.

Набережночелнинский институт ФГАОУ ВО Казанский (Приволжский) федеральный университет, 423810, г. Набережные Челны, пр. Мира, д.68/19

e-mail: daria_013@mail.ru

поступила в редакцию 23 января 2017 года

Аннотация

Описывается онтологическая модель технических требований к этапу эксплуатации автомобиля «КАМАЗ» и показывается принципиальная возможность описания других этапов.

Ключевые слова: *онтология, технические требования, жизненный цикл, автомобиль «КАМАЗ».*

Введение. В настоящее время существует проблема того, что информация по техническим требованиям разнесена по разным источникам. Для того чтобы получить и систематизировать необходимую информацию, порой приходится рассмотреть десятки книг, справочников, ГОСТов. Такой подход занимает большое количество времени. Таким образом, встает важная задача - составление базы знаний для экспертной системы, которая позволяла бы работникам автомобильной отрасли получать сведения о технических требованиях к некоторым этапам жизненного цикла автомобиля.

Одним из перспективных способов представления знаний является онтология.

Онтология – это результат формализации некоторой области знаний с помощью концептуальной схемы. Концептуализацией именуется абстрактное упрощенное представление мира, которое формируется для некоторых целей [1]. Иными словами, онтология - это система, состоящая из понятий и набора утверждений об этих понятиях с целью строить классы, объекты, отношения, функции и теории. Эта формализация способствует такому пониманию предметной области, с помощью которого могут общаться и люди, и компьютерные системы [2].

Целью создания онтологий является обеспечение поддержки работы по накоплению, разделению и вторичному использованию знаний.

Основная часть. Онтология состоит из:

- индивидов – это конкретные объекты интересующей предметной области. Индивиды иногда называют экземплярами;
- свойств – это бинарные отношения на индивидах. Другими словами, это свойства, которые соединяют двух индивидов;
- классов – множества, элементами которых являются индивиды. Классы описываются с использованием формальных конструкций, которые декларируют требования для членства в классе.

Для создания, редактирования, а также поддержки онтологий существует целый ряд инструментального программного обеспечения. Многие из этих инструментов позволяют создавать и редактировать онтологию, поддерживать визуализацию, импорт и экспорт для онтологий различных форматов, а также документирование. Часто подобные редакторы выражают онтологию на одном из многих языков онтологий [3].

Для разработки онтологии был выбран редактор Protégé [4]. В отличие от других редакторов, Protégé - это открытый, свободно распространяемый редактор онтологий. Protégé предоставляет графический пользовательский интерфейс для построения онтологий и предоставляет возможность графического представления онтологии. Также Protégé поддерживает открытый

исходный код онтологии на OWL. Редактор онтологий Protégé позволяет проектировать онтологии, разворачивая иерархическую структуру абстрактных или конкретных классов. Таким образом, структура онтологии подобна иерархической структуре каталога. Кроме того, Protégé снабжен большим количеством примеров онтологий. Более того, данная программа имеет возможность поддержки модулей расширения функциональности. Protégé доступен для свободного скачивания с официального сайта вместе с плагинами и примерами онтологий.

Была выбрана версия Protégé 5.0. На момент проведения моих исследований данная версия была самой новой. В Protégé 5.0 были исправлены и учтены недоработки прошлых версий.

Разработка онтологии. Для описания конкретных объектов были выделены индивиды (рисунок 1).

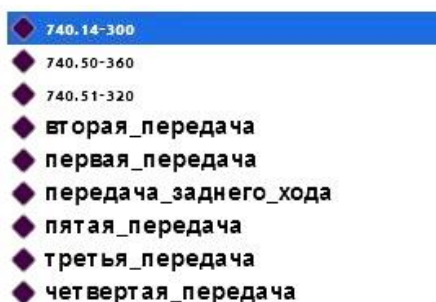


Рисунок 1. – Индивиды, представленные в онтологии.

Каждый из индивидов может иметь одно или несколько свойств, а свойство может принимать конкретное значение. Например, первые три элемента представляют собой названия моделей двигателей автомобиля КАМАЗ. Они обладают свойством *частота_вращения_коленчатого_вала*, которое принимает значение 2600 для индивида *740.14-300*.

Имеет смысл обозначить индивиды, которые отражают характеристики коробки передач: первая, вторая и т. д. Эти индивиды обладают свойством *суммарный_угловой_зазор_в_КП_вторичного_вала*. Например, индивид *первая_передача* имеет свойство *суммарный_угловой_зазор_в_КП_вторичного_вала* со значением [>3].

Для разрабатываемой онтологической модели были определены свойства классов, показывающие бинарные отношения между классами (таблица 1).

Таблица 1. – Свойства классов.

| | |
|-----------------|--|
| влияет_на | определяет отношение между этапами жизненного цикла и требованиями к нему; |
| зависит_от | определяет отношение между требованиями к автомобилю и его системами; |
| ссылка_на | позволяет сослаться на документ, который дает определение терминологии класса; |
| состоит_из | показывает, из каких деталей состоит выбранная часть автомобиля; |
| является_частью | показывает, частью какой детали является выбранный элемент. |

Под полным жизненным циклом изделия понимают время (и соответствующие ему действия), которое проходит от постановки четкой задачи создания этого изделия (автомобиля и др.) до его полного физического или морального износа и утилизации.

В полный жизненный цикл автомобиля входят следующие стадии:

1. Разработка требований
2. Создание функциональной модели
3. Проектирование автомобиля в CAD-системе
4. Виртуальные испытания
5. Планирование производства
6. Моделирование технологических процессов
7. Испытание автомобиля

8. Создание прототипа для ОПП
9. Выпуск автомобиля
10. Продажа
11. Эксплуатация
12. Утилизация

Стадии полного жизненного цикла автомобиля были внесены в онтологическую модель в качестве классов (рисунок 2).

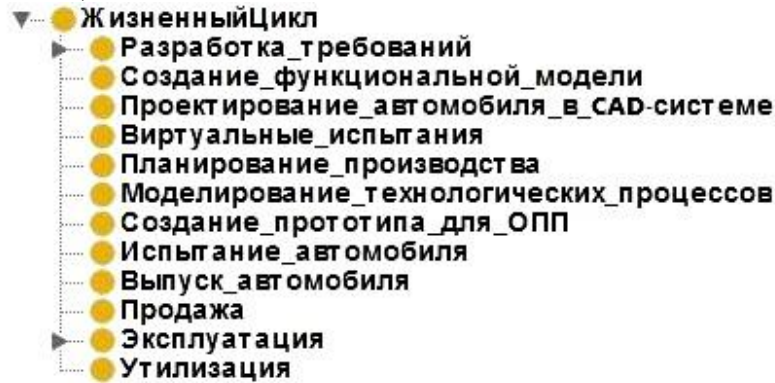


Рисунок 2. – Стадии полного жизненного цикла автомобиля.

В класс *Разработка_требований* были включены названия основных технических требований по полному жизненному циклу автомобилей: надежность, безотказность, долговечность, отказоустойчивость, ремонтпригодность и сохраняемость. Были даны определения приведенным требованиям. Кроме того, в класс *Разработка_требований* было добавлено обозначение ГОСТов (рисунок 3).



Рисунок 3. – Подклассы класса «Разработка_требований».

Также была рассмотрена зависимость требований к автомобилям от стадий жизненного цикла. Например, рисунок 4 иллюстрирует отношение требования *надежность* к разным этапам жизненного цикла.

Рассмотреть жизненный цикл полностью с раскрытием всех его стадий в рамках данной работы не представляется возможным. Важно было показать принципиальную возможность создания онтологического описания технических требований по полному жизненному циклу автомобиля на примере этапа эксплуатации. Этот этап, как правило, является наиболее длительным из всех этапов жизненного цикла.

Этап эксплуатации при разработке онтологии был выделен в отдельный класс. Узлы и агрегаты автомобиля, его эксплуатационные материалы были выделены в подклассы так называемого суперкласса *Эксплуатация* (рисунок 5).

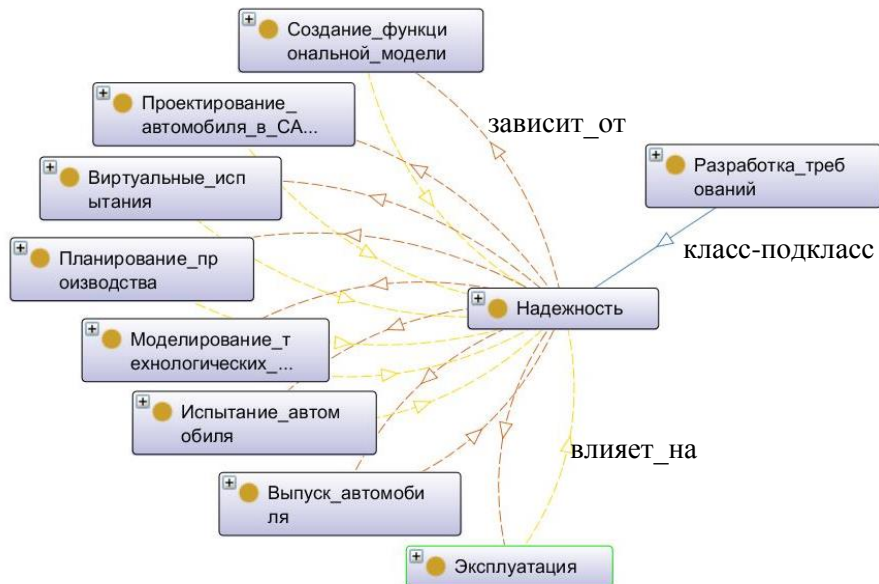


Рисунок 4. – Зависимость *надежности* от стадий жизненного цикла.

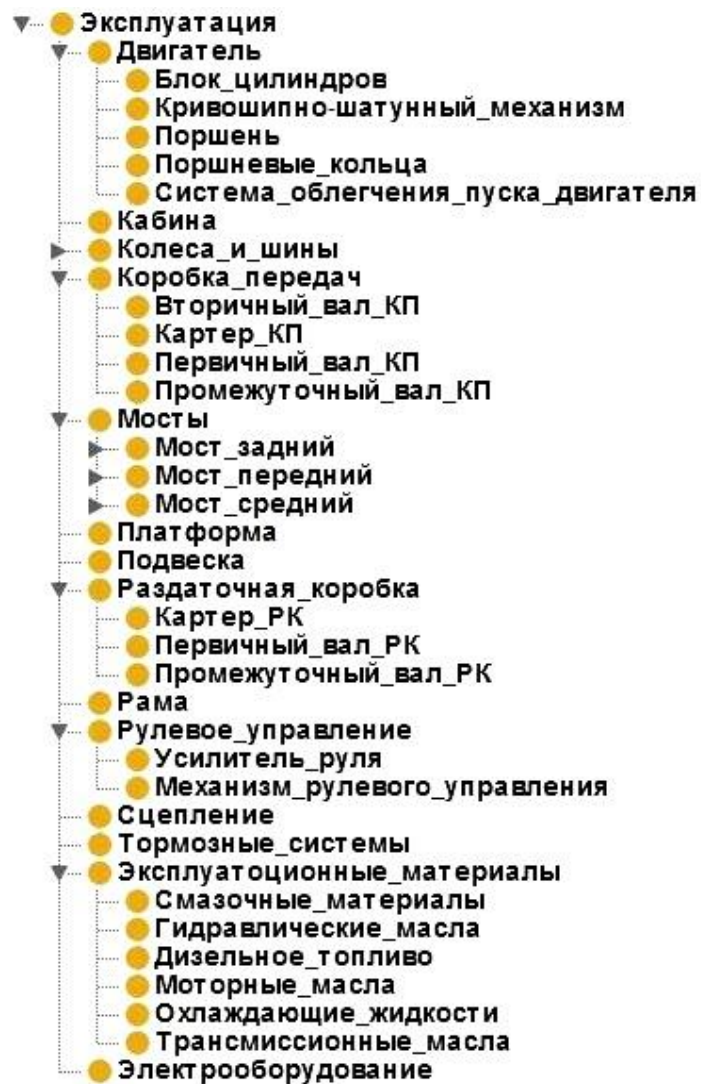


Рисунок 5. – Подклассы класса *Эксплуатация*.

Наиболее подробно был описан класс *Мосты* (рисунок 6).

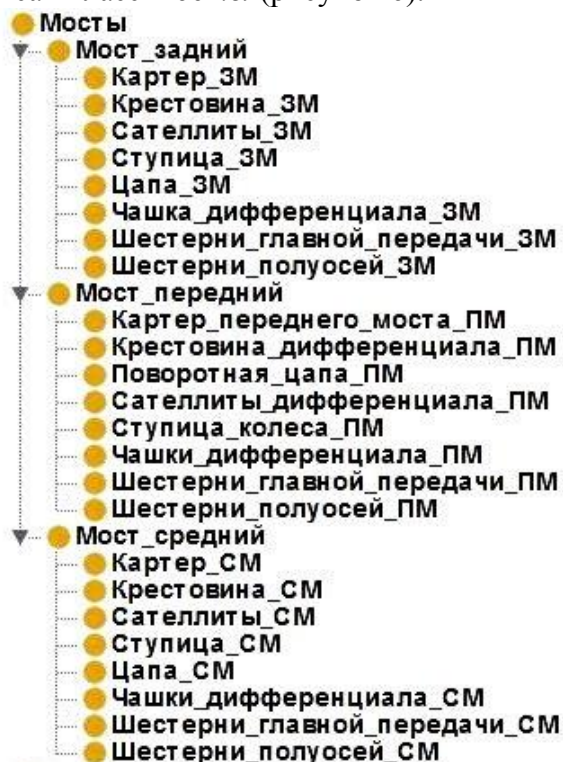


Рисунок 6. – Подклассы класса «Мосты».

В данной работе были созданы:

- 91 класс;
- 9 экземпляров;
- 8 свойств.

Моя база знаний позволяет работнику ПАО «КАМАЗ» извлекать сведения о технических требованиях к определенным узлам автомобиля на этапе эксплуатации. Делается это с помощью запросов. Для создания запросов к онтологии использовался язык SPARQL[5].

Рисунок 7 показывает запрос, который выводит подклассы класса *Разработка_требований*, связанные с этим классом свойством *является_частью*. На этом рисунке виден результат выполнения этого запроса.

SPARQL query: ⌵ ⌵ ⌵

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX : <http://www.semanticweb.org/пользователь/ontologies/2016/4/untitled-ontology-3#>
SELECT ?Разработка_требований
WHERE {
  ?Разработка_требований rdfs:subClassOf ?object.
  ?object owl:onProperty :является_частью.
  ?object owl:someValuesFrom :Разработка_требований.
}
        
```

| Разработка_требований |
|-----------------------|
| Безотказность |
| Долговечность |
| Надежность |
| Отказоустойчивость |
| Ремонтопригодность |
| Сохраняемость |

Рисунок 7. – Пример запроса.

Заключение. Были определены свойства классов, соответствующие этапу эксплуатации, сами классы онтологии и их дерево иерархии, а также индивиды. В качестве инструмента бала выбрана программа Protégé. Были реализованы SPARQL-запросы.

Благодарность. Выражаю благодарность своему научному руководителю Товштейну Марку Яковлевичу, Карабцеву Владимиру Сергеевичу, который выступил в роли эксперта при разработке базы знаний, а также Демьянову Дмитрию Николаевичу за помощь в подготовке данной статьи.

Список литературы

- 1) Тузовский А.Ф., Чириков С.В., Ямпольский В.З. Системы управления знаниями (методы и технологии) / Под общ. ред. В.З. Ямпольского. Томск: Изд-во НТЛ, 2005. 260 с.
- 2) Товштейн М.Я. Возможность применения онтологической модели автомобиля в базе знаний / М.Я. Товштейн, Л.И. Сунгатуллин // Социально-экономические и технические системы: исследование, проектирование, оптимизация. - Набережные Челны: Набережночелнинский институт (филиал) Казанского (Приволжского) федерального университета. 2016. №2 (69). С.52-62
- 3) Овдей О.М., Проскудина Г.Ю. Обзор инструментов инженерии онтологий. Российский научный электронный журнал «Электронные библиотеки», 2004, Выпуск 4, <http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2004/part4/op> (Дата обращения: 20.06.2016)
- 4) Муромцев Д.И. Онтологический инжиниринг знаний в системе Protégé: Методическое пособие. — СПб: СПбГУ ИТМО, 2007. 62 с. http://window.edu.ru/window_catalog/redirect?id=54429&file=itmo240.pdf (Дата обращения: 17.06.2016)
- 5) Прудхоммоукс Эрик, Сиборн Энди. SPARQL Query Language for RDF W3C Recommendation 15 January 2008 // SPARQL Query Language for RDF: URL: <http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/> (Дата обращения: 18.06.2016)