



Е.М. Лаврищева, Л.Е. Карпов,
А.Н. Томилин

**Семантические ресурсы для
разработки онтологии научной и
инженерной предметных областей**

Рекомендуемая форма библиографической ссылки

Лаврищева Е.М., Карпов Л.Е., Томилин А.Н. Семантические ресурсы для разработки онтологии научной и инженерной предметных областей // Научный сервис в сети Интернет: труды XVIII Всероссийской научной конференции (19-24 сентября 2016 г., г. Новороссийск). — М.: ИПМ им. М.В.Келдыша, 2016. — С. 223-239. — doi:[10.20948/abrau-2016-16](https://doi.org/10.20948/abrau-2016-16)

Размещена также [презентация к докладу](#)

Семантические ресурсы для разработки онтологии научной и инженерной предметных областей

Е. М. Лаврищева¹, Л. Е. Карпов², А. Н. Томилин³

¹Е. М. Лаврищева <lavr@ispras.ru>

*Институт системного программирования РАН,
Московский физико-технический институт (технический университет),*

²Л. Е. Карпов <mak@ispras.ru>

*Институт системного программирования РАН,
Московский государственный университет им. М. В. Ломоносова,*

³А. Н. Томилин <tom11@bk.ru>

*Институт системного программирования РАН,
Московский государственный университет им. М. В. Ломоносова*

Аннотация. Представлены основные положения и ресурсы онтологии семантической сети для определения концептуальных моделей сложных предметных областей и сетевых приложений. Ресурсами являются предметно-ориентированные языки описания онтологий (OWL, DSL и др.) и системы их поддержки (FODA, Protégé, DSL Tool VS.Net и др.). Дана характеристика основных понятий научной области – вычислительная геометрия и инженерного домена программной инженерии – жизненный цикл стандарта ISO/IEC 12207. Представлен общий набор семантических ресурсов онтологии, включая средства предметно-ориентированного языка DSL и систем Protégé и DSL Tool VS.Net для представления этих предметных областей. Приведены примеры их описания.

Ключевые слова: семантическая сеть; онтология; модель онтологии, ресурсы онтологии, предметный язык, концептуальная модель, OWL, Protégé; бизнес приложения, задачи

1. Введение

Семантик-WEB – эта новая мировая информационная среда глобальной информационной сети (World Wide Web), которая содержит семантические ресурсы, языки и инструменты разработки онтологий, систем и бизнес-процессов с использованием накопленных знаний. Термин семантическая сеть (semantic-WEB) ввел Тим Бернес-Ли (2001) в журнале «Scientific American», где им были определены семантические основы глобальной сети. С тех пор это понятие развивается исследовательскими программами институтов США и Евросоюза, множеством больших и малых производителей систем, проектами с

открытым кодом, а также конференциями по семантическим ресурсам и технологиям (<http://semwebprogramming.org>).

Семантическая сеть предоставляет научный сервис для постановки и автоматизированной обработки научных задач, больших данных в разных форматах, интеграцию данных из коллажей (Mash-Ups), поиск и композицию сетевых служб, управление интеллектуальными агентами в мобильных приложениях и др. Она обеспечивает решение задач с использованием многочисленных сервисов и научных достижений в области предоставления бизнес-услуг, предлагает новые подходы к использованию больших объемов информации, основанных на новых словарях и концептуальных моделях, методах онтологизации новых научных задач и приложений с помощью предметно-ориентированных языков (OWL, DSL и др.) и инструментов (FODA, Protégé, DSL Tool VS.Net и др.), см., например, [1-13].

Данная работа посвящена описанию языков и инструментов онтологии в глобальной информационной сети для представления научного домена «Вычислительная геометрия» и инженерного домена «Жизненный цикл разработки программных систем» в глобальной семантической среде [14-25].

2. Базовые ресурсы семантического веба

Глобальная семантическая сеть включает в себя множество ресурсов и инструментов (рис.1) [1].

Языки семантической сети включают набор ключевых слов, которые позволяют описывать компоненты и утверждения. Применяются такие языки для описания онтологий, бизнес процессов и данных (OWL, RDF, BPMN и др.), а также для задания уникальных идентификаторов ресурсов – Uniform Resource Identifier (URI), Uniform Resource Locator (URL) и др. Язык URL задает уникальные имена для всех элементов глобальной сети. С его помощью задается расширяемое пространство имен. Доступ к различным уровням памяти, адреса которых содержатся в URL, осуществляется с помощью уникальных имен ресурсов URN (Uniform Resource Name).

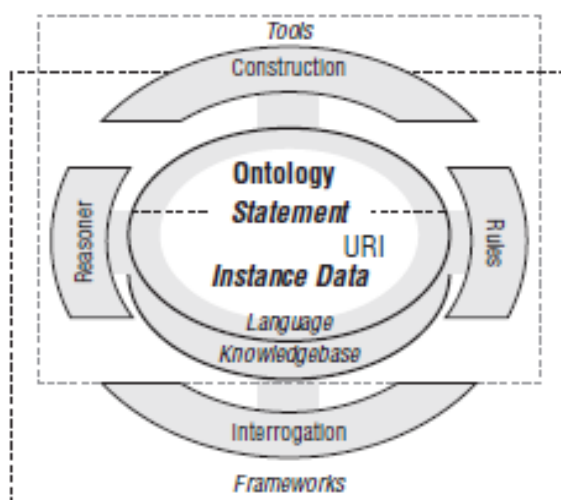


Рис. 1. Основные компоненты глобальной семантической сети

Онтология определяет понятия, отношения и ограничения объектов в концептуальной модели предметной области. Многие онтологии могут быть включены в любое приложение путем адаптации к специфическим потребностям этой области. Онтология поддерживает коммуникации между приложениями и является способом описания доменов таких областей, как финансы, медицина, экономика, бизнес приложения и др.

Инструменты могут относиться к одному из четырех типов: конструирования и развития приложений семантической сети, справочные инструменты для изучения сети, инструменты-резонеры, добавляющие механизмы вывода и машины правил для расширения семантической сети.

Инструменты конструирования позволяют конструировать или интегрировать семантическую сеть путем создания или импорта компонентов для онтологии экземпляров. Некоторые графические инструменты (GUI) позволяют осуществлять просмотр и исследование данных сети, образуя полезный редактор семантической сети.

Справочные инструменты обеспечивают навигацию по семантической сети в поисках ответа на вопрос пользователя. Существуют различные справочные методы, начиная от простой навигации по графу при поиске и до полного применения языка запросов.

Механизмы-резонеры добавляют к семантической сети новые понятия, необходимые пользователям. Компоненты создают логические дополнения путем классификации. Классификация заполняет структуру класса, позволяя должным образом соотносить понятия и отношения с другими классами. Имеется несколько типов резонеров, предлагающих различные уровни рассуждений. Резонеры включаются в другие инструменты и каркасы. Они служат рычагами для создания логически правильных вспомогательных утверждений.

Машины на правилах – это машины вывода, основанные на правилах дескрипторной логики. Они добавляют измерение конструкций знаний. Правила позволяют выполнять слияние онтологий и других логических задач, в том числе поиск с подсчетом и по строке (count and string searches). Машины на правилах могут рассматриваться как часть общего представления знаний. Каждая машина правил придерживается заданным языком описания правил.

Семантический каркас объединяет перечисленные выше инструменты и позволяет им работать как единое целое. Утверждения, универсальные идентификаторы (URI), языки, онтологии, и данные конкретных экземпляров классов составляют семантически связанную информацию в семантической сети, которой манипулирует семантический каркас, создавая новые инструменты, обогащая эту сеть.

Сетевые данные отражают смысловое значение данных и интеграции данных для совместного использования путем доступа и обмена богатыми информационными ресурсами глобальной информационной сети, включая использование многих существующих источников данных.

Динамические данные сети позволяют добиваться динамичного (во время выполнения) изменения структуры и содержания информации.

Фреймворки Семантической Сети — это набор программных средств, сред, библиотек для создания, манипулирования и обогащения семантической сети. Они помогают создавать выражения, онтологии, сетевые приложения и публиковать их в глобальной сети. Примерами таких средств и сред могут служить среда программирования Eclipse IDE, язык программирования Java, набор библиотек для работы с Apache Jena и средств создания онтологии Protégé.

Язык **RDF** был утвержден в качестве стандарта консорциумом W3C в 2004 году. Он предназначен для систематического описания сетевых ресурсов, понятых компьютеру. Формат RDF предназначен для сохранения метаданных (данные о данных), описания семантических ресурсов, то есть служит каркасом для создания отдельных компонентов глобальной семантической сети. Документы в RDF обрабатываются компьютером автоматически. Схема RDF, обозначаемая часто как RDFS (от английского *RDF Schema*) — это надстройка над RDF, которая позволяет создавать классы и свойства объектов.

Язык **OWL** (Web Ontology Language) используется с 2004 года, он построен на форматах RDF и RDFS и предназначен для обработки информации в сети. Язык OWL имеет 3 степени детализации, легко масштабируется и согласовывается с современными сетевыми стандартами. В 2008 году был принят новый стандарт OWL 2, включающий описание логики.

Язык **SPARQL** (Protocol And RDF Query Language) — это новый язык запросов для быстрого доступа к данным RDF. Используя обычный протокол и язык SPARQL, программы могут анализировать RDF-описания ресурсов и получать из сети необходимую информацию.

В качестве формата обмена правилами, наряду с другими форматами рекомендуется использовать формат **RIF** (Rule Interchange Format).

Онтологии для представления накопленных знаний

Онтология является основой стандартизации мировой системы знаний, в том числе языковой, инженерной, системной деятельности. Появились международные профильные стандарты терминов и определений и ряд международных органов, которые ответственные за их ведение (ISO, W3C и некоторые другие).

В основе представления любой системы знаний лежит понятийная база, совокупность концептов (понятий) и отношений между ними, классификация понятий и их таксономия в виде тезаурусов, а также методы создания профильных онтологий.

- онтология Sensus базируется на понятиях естественного (английского) языка и содержит более чем 70 000 терминов и их дефиниций;
- онтология понятий электронной коммерции;

- глобальная онтология продуктов и услуг (стандарт Объединенных Наций);
- коммерческая онтология SCTG транспортных потоков товаров компаний;
- онтология e-cl@ss поддерживает обмен данными и материалами между продавцами и пользователями крупных компаний Германии;
- онтология товаров RosettaNet поддерживается более, чем 400 коммерческими компаниями.

Существует и широко используется круг медицинских онтологий (Galen для определения клинической картины; UMLS для Национальной медицинской библиотеки Соединенных Штатов; ON9 для аттестации известных медицинских систем по определенным параметрам). Одновременно существуют также инженерные онтологии; онтологии деятельности предприятий; химические и биологические онтологии и многие другие.

К основным направлениям создания онтологий относятся:

- разработка онтологических моделей, методов и средств доступа к мировым информационным ресурсам;
- создание моделей описания готовых ресурсов разработки как повторно используемых знаний;
- создание библиотек онтологий по профилям знаний;
- разработка методов и средств создания комплекса инструментов ведения онтологий;
- разработка методических и учебных материалов по применению семантической сети для описания профессиональных знаний.

Онтология в семантической глобальной сети дает аппарат построения концептуальной модели некоторой предметной области, которая включает понятия, отношения и ограничения ее элементов. Средствами уже перечисленных языков семантической сети формализуется любая предметная область из накопленных знаний. Концептуальная модель включает в себя структуры данных, содержащие релевантные классы объектов, их связи и правила (теоремы, ограничения), принятые в этой области (рис.2) [1].

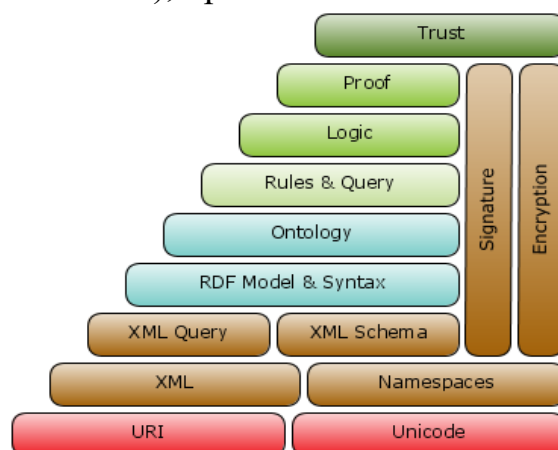


Рис.2. Набор понятий семантической сети

При описании онтологии некоторой предметной области используются индивиды, понятия, атрибуты и отношения.

Индивиды – это основные, низкоуровневые компоненты онтологии. Индивиды могут представлять собой как физические объекты (люди, дома, планеты), так и абстрактные построения (числа, слова).

Понятия – абстрактные группы, коллекции или наборы объектов, которые включают в себя индивиды, классы или же их сочетания.

Объекты могут иметь атрибуты. Каждый атрибут имеет имя и значение, которые используются для сохранения информации, специфической для каждого объекта.

Описание модели предметной области выполняется с помощью языков OWL, RDF, RDFS и KIF (Knowledge Interchange Format) [2, 3], а также логических языков. Описание модели состоит из S-выражений логики и других аналогичных форм. Для работы с языками онтологий применяются самые разные программные инструменты: редакторы онтологий (для создания онтологий), системы управления базами данных (DBMS) онтологий (для хранения и обращения к онтологии) и хранилища онтологий (для работы с разными онтологиями).

Описание онтологии на языке OWL – это последовательность аксиом и фактов, информации о классах, свойствах, ресурсах и документах веб, которые импортируются через URI и могут ссылаться на XML–схему по форме:

< Datatype ID >:: = < URI ссылки >.

3. Онтологические элементы для задания доменов

К языкам описания предметной области (домена) относятся: языки DSL (Domain Specific Language), FODA (Feature-Oriented Domain Analysis [5]), DSSA (Domain-Specific Software Architectures) и инструменты реализации DSL – Eclipse-DSL [6], а также ODSD (Ontology-Driven Software Development), который позволяет задавать описания классов, отражающих понятия домена. Одним из основных средств представления конкретной онтологии некоторой предметной области является система Protégé [6-11].

К средствам описания онтологий доменов относятся метапонятия – классы, факты, аксиомы, фасеты, слоты и другие аналогичные.

Классы описывают понятия предметной области, а *слоты* – свойства (атрибуты) классов. *Фасеты* описывают свойства слотов (конкретные типы и возможные диапазоны значений). Класс может быть эквивалентом, подмножеством или пересечением более общих классов с ограничениями. *Аксиома* класса содержит совокупность описаний, которые могут иметь вид обобщенных классов, ограничений, наборов ресурсов, булевых комбинаций описаний.

Аксиома содержит совокупность понятий, которые могут относиться к обобщенным классам и включать ограничения, наборы ресурсов, булевы комбинации описаний и прочее. Каждая аксиома задается с помощью правил и

ограничений. Абстрактные классы являются контейнерами конкретных классов и могут содержать абстрактные атрибуты (которые не содержат конкретных значений). Атрибуты понятий предметной области являются слотами, образующими классы, которыми могут быть значения (экземпляры атрибутов). Согласно фреймовой модели представления знаний, которая используется в Protégé, слот является фреймом. Слоты определяются независимо от какого бы то ни было класса, и один и тот же слот может принадлежать разным классам.

Фасеты позволяют вводить ограничение на типы данных и диапазоны значений их экземпляров (значений атрибутов), подобно понятиям XML-схемы. Слот определяется количеством значений в экземпляре и ограничениями для типов этих значений (например, целое, символьное и т. п.) и граничными значениями (min, max). Фасеты задают ограничение на присоединение слота к фрейму класса. Слоты-образцы (template slots) и собственные (own) слоты можно присоединить к фрейму (класса) одним из двух способов: как слот-образец или как собственный слот, который присоединяется к фрейму. Классы также могут иметь собственные слоты. Например, документация класса является собственным слотом, присоединенным к классу, поскольку описывает сам класс, а не экземпляры класса.

Факты — это информация о специфических ресурсах в форме классов и свойств со значением этого ресурса. Аксиомы используются, чтобы сопоставить класс и свойство идентификации с частичными или полными спецификациями характеристик, поставщики могут предлагать другую логическую информацию о классах и свойствах. Каждая аксиома класса содержит совокупность более общих классов и совокупность локальных ограничений свойства.

Элементы онтологий могут вступать в следующие отношения:

- обобщение для сужения существенных признаков понятия с расширением круга понятий объектов и их объема;
- конкретизация, как добавление существенных признаков, благодаря чему содержание понятия расширяется, а объем его сужается;
- агрегация, как объединение ряда понятий в новое понятие, существенные признаки нового понятия могут задаваться суммой признаков;
- ассоциация, как наиболее общее отношение, которое утверждает наличие связи между понятиями, не уточняя зависимости их от содержания и объемов.

3.1. Описание домена – Вычислительная геометрия

Вычислительная геометрия (Computational Geometry) [14-18] – отрасль компьютерных наук для изучения алгоритмов в терминах геометрии. Основным стимулом развития вычислительной геометрии как дисциплины обучения был прогресс в компьютерной графике и в системах автоматизированного

проектирования. Многие задачи вычислительной геометрии являются классическими, преподаются в различных университетах мира и могут появляться при математической визуализации.

Другим важным применением вычислительной геометрии является робототехника (планирование движения роботов и задачи распознавания образов), геоинформационные системы (геометрический поиск, планирование маршрута), дизайн микросхем, программирование станков с числовым программным управлением и др.

Основными разделами вычислительной геометрии являются:

- комбинаторная вычислительная геометрия или алгоритмическая геометрия, которая рассматривает геометрические объекты как дискретные сущности. основополагающим материалом является книга Препарати и Шеймоса, в которой в 1975 году впервые был использован термин "вычислительная геометрия" [14].
- численная вычислительная геометрия или машинная геометрия, геометрическое моделирование, которое имеет дело в основном с представлением объектов реального мира в форме пригодной для дальнейшей компьютерной обработки (1971 год). Этот раздел можно рассматривать как дальнейшее развитие вычислительной геометрии и часто рассматривается как раздел компьютерной графики.

Комбинаторная вычислительная геометрия содержит разработанные эффективные алгоритмы и структуры данных для решения задач, которые заданы в терминах базовых геометрических объектов: точек, отрезков, многоугольников, многогранников и других.

Вычислительная геометрия предназначена для работы над очень большими наборами данных с десятками или сотнями миллионов точек. Основные задачи вычислительной геометрии можно классифицировать разными способами и с разными критериями.

Пример задачи. На множестве n точек на плоскости найти пару точек, расстояние между которыми наименьшее. Расстояние между каждой парой точек вычисляется по формуле $n(n-1)/2$. Затем выбирается пара с наименьшим расстоянием. Полный перебор имеет сложность $O(n^2)$, то есть время его выполнения пропорционально квадрату количества точек. Имеется также алгоритм со сложностью $O(n \cdot \log n)$.

В вычислительной геометрии различают статические задачи и задачи геометрического поиска.

Статические задачи. В задачах этой категории на вход подаются некоторые данные, за ними алгоритм должен вычислить соответствующие результаты. К фундаментальным задачам такого рода относятся:

- выпуклая оболочка: имея набор точек необходимо найти наименьший выпуклый многоугольник, который содержит все точки;

- пересечение отрезков: найти все пересечения в наборе отрезков;
- триангуляция Делона;
- диаграмма Вороного: имеется множество точек на плоскости, требуется разделить это множество на области, чтобы каждая такая область образовывала множество точек, более близких к одному из элементов исходного множества, чем к любому другому его элементу;
- задача ближайшей пары точек: имея набор точек найти кратчайшее расстояние;
- евклидов кратчайший путь: соединить две точки Евклидова пространства (с полигональными препятствиями) кратчайшим образом.
- триангуляция многоугольника: имея многоугольник, разбить его на внутренние треугольники.

Задачи геометрического поиска (запроса). В задачах геометрического поиска входные данные состоят из двух частей: пространств поиска и запросов, которые разнятся в различных видах задач. Для обеспечения эффективного выполнения нескольких запросов пространство поиска требует предварительной обработки:

- региональный поиск: обработать набор точек, с целью эффективного поиска набора точек, содержащихся в заданном регионе.
- локализация точки: имея разбиение пространства на регионы, создать структуру данных, которая позволит эффективно определить, в каком регионе находится данная точка.
- поиск ближайшего соседа: обработать набор точек чтобы иметь возможность эффективно найти точки, близкие к заданной.
- трассировка лучей: для заданного набора объектов в пространстве создать структуру данных, которая позволит эффективно определять объекты, пересекающие заданный луч.

Динамические задачи – это тип задач, входные данные которых постепенно меняются (например, добавляются или удаляются объекты). Алгоритмы решения таких задач включают в себя поддержку динамических структур данных. Любую задачу вычислительной геометрии можно решать динамично, но за счет дополнительных вычислительных ресурсов. Региональный поиск, построение выпуклой оболочки можно проводить над множеством точек, которые меняются.

Вычислительная сложность для этого класса задач задается параметрами, к которым относятся:

- ресурсы, необходимые для определения структур данных при поиске;

- ресурсы, необходимыми для модификации построенной структуры и задания ресурсов для ответа на запросы.

Некоторые задачи могут рассматриваться как такие, которые принадлежат нескольким категориям в зависимости от контекста.

Вариации. Во многих программах эта задача рассматривается как задача первого класса. Тем не менее, во многих случаях нужно определить положение курсора «мышь» внутри данного многоугольника. Курсор постоянно перемещается, а многоугольник не меняется. Аналогично можно проверять изображенный на экране радара подозрительный летательный аппарат, не пересек ли он охраняемую границу страны. Такие задачи можно считать задачами геометрического запроса. В САД-системах сам многоугольник может варьироваться, поэтому задача может считаться динамической. Таким образом в описание онтологической схемы комбинаторной вычислительной геометрии входят (рис.3):

1. Классы задач
 - 1.1. Статистические задачи.
 - 1.2. Задачи геометрического поиска.
 - 1.3. Динамические задачи.
 - 1.4. Вариации.

Реализация онтологии «Вычислительная геометрия». Онтология домена «Вычислительная геометрия» для заданного выше описания выполнена в системе Protégé 4.1 (рис.3), которая является свободно распространяемым открытым редактором онтологий и фреймворком для построения баз знаний предметной области.

Платформа Protégé поддерживает два основных способа моделирования онтологий с помощью редактора Protégé -Frames и Protégé -OWL. Онтологии, построенные в Protégé, могут быть экспортированы в другие форматы, включая RDF (RDF Schema), OWL и XML Schema.

Платформа Protégé поддерживается значительным сообществом, состоящим из разработчиков и ученых, правительственных и корпоративных пользователей, которые используют его для решения задач, связанных со знаниями, в таких разнообразных областях, как биомедицина, сбор знаний и корпоративное моделирование. Онтология домена «Вычислительная геометрия» включает набор понятий и связей между ними, которые зафиксированы в схеме онтологии (рис.3) и в XML. Кроме того, в Киевском национальном университете (КНУ) реализован электронный учебник для обучения <http://cg.incyb.kiev.ua>, в котором описаны алгоритмы геометрии (см. [16]).

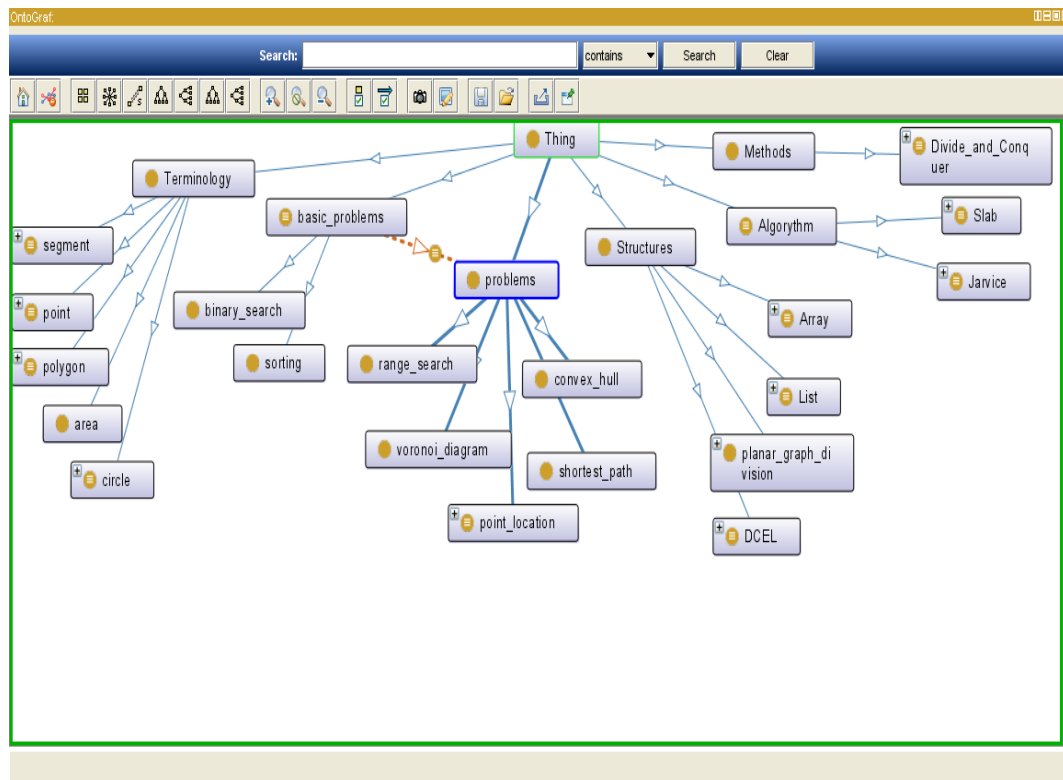


Рис.3. Онтология вычислительной геометрии

3.2. Онтология домена — стандарт жизненного цикла 12207

При проектировании разного рода прикладных систем используются модели жизненного цикла или стандарт жизненного цикла программного обеспечения ISO/IEC 12207-ЖЦ. Стандарт жизненного цикла является общим механизмом регламентированного построения разных программных систем. Жизненный цикл в данном стандарте представлен тремя категориями процессов (рис.4, 5) [20-25]:

- 1) основные процессы;
- 2) процессы поддержки;
- 3) организационные процессы.

Для каждого из процессов определены виды деятельности (действия – activities), задачи (tasks), совокупность результатов (выходов) деятельности для решения задач и др. В стандарте приведен перечень работ для этих процессов, но не задан способ их выполнения и форм представления результатов. Основные процессы – разработки, эксплуатации и сопровождения программных систем (рис. 4).

Стандарт жизненного цикла содержит в себе также вспомогательные процессы, которые регламентируют дополнительные действия по проверке продукта, управлению проектом и качеством (рис. 5). Как правило, в зависимости от целей конкретного проекта на ПП Главный разработчик и

менеджер выбирают процессы, действия и задачи, выстраивают определенную модель жизненного цикла для этого проекта.

Этот стандарт (в варианте 2007 года) включает в себя (таблица 1) 17 процессов, 74 подпроцесса и 232 технологические операционные задачи (действия). Этого набора процессов и задач необходимо и достаточно для проектирования систем с помощью процессного подхода. Некоторые системные фирмы поддержки реализуют отдельные фрагменты или варианты этого стандарта.

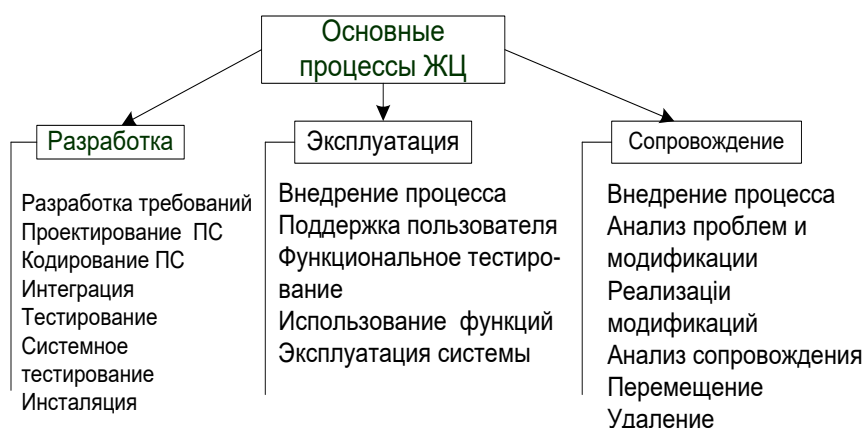


Рис. 4. Схема основных процессов жизненного цикла программной системы для онтологии



Рис. 5. Схема вспомогательных процессов (процессов поддержки и организационных процессов) жизненного цикла программной системы

Таблица 1. Процессы, подпроцессы и задачи жизненного цикла

| Классы | Процесс | Действие | Задача |
|--------------------------|---------|----------|--------|
| Основные процессы | 5 | 35 | 135 |
| Процессы поддержки | 8 | 25 | 70 |
| Организационные процессы | 4 | 14 | 27 |
| Всего | 17 | 74 | 232 |

Концепция автоматизации стандарта жизненного цикла средствами онтологии является новой. В основе реализации онтологии лежит структура процессов жизненного цикла, их взаимосвязи по передаваемым результатам проектирования программной системы, а также функции процессов для их реализации и выполнения. Если все 17 процессов будут реализованы, то можно из них генерировать некоторые подмножества жизненного цикла, как вариант рабочего жизненного цикла для конкретного применения [20].

Имеются разнообразные языковые и технологические средства формального описания процессов жизненного цикла для последующего автоматизированного моделирования разных программных продуктов. К такого рода средствам относятся нотация BPMN для описания процессов жизненного цикла и язык DSL для описания семантики доменов. В качестве примера реализации процессов жизненного цикла избран онтологический подход. В нем жизненный цикл представляется с помощью словарей понятий, концептов и отношений между ними в среде Protégé, DSL Tool VS.Net и др. Онтологическое описание трансформируется к языку XML, который является языком реализации размеченных данных домена жизненного цикла, установленных связей и обменов данными между процессами.

Концепция онтологизации жизненного цикла обсуждалась в КНУ на научных семинарах кафедры теории программирования и технологии, кафедры информационных систем (ИС) и МФТИ в группах студентов, которым читаются лекции по предмету "Программная инженерия".

Домен жизненного цикла занимает центральное место в программной инженерии, основным назначением которого есть методы и средства изготовления сложных программных систем. Студенты изучали эти методы и средства, а также современные стандарты жизненного цикла ISO/IEC 12207–2007 и ISO/IEC 11404–2006. GDT (General Data Types). С участием студентов был разработан экспериментальный вариант онтологии ЖЦ с помощью открытых инструментов – DSL Tools VS.Net и Protégé [4–6].

В каждой технологии программирования сложных программных систем с использованием стандарта жизненного цикла применяются теоретические, прикладные методы, стандарты качества, общие и фундаментальные типы данных (ISO/IEC 15404, ISO/IEC 9126, ISO/IEC 11404 GDT и др.), а также методики этих стандартов. Участники разработки жизненного цикла изучили современные онтологические средства и средства визуального представления процессов жизненного цикла – DSL Tools VS.Net, Protégé и др. На основе этих средств было получено первое описание онтологии жизненного цикла в графическом виде и в XML. На конференции "Science and Information -2015" по онтологии жизненного цикла было предложено запатентовать эту идею [21].

Для представления онтологической структуры жизненного цикла (рис.3) используется графический язык DSL, который содержит общие абстракции для отображения классов объектов предметной области, типов процессов и

действий, а также отношений между ними. Описание в этом языке сводится к языкам HTML, XML, WSDL и др.

Модель предметной области жизненного цикла описана на одном из языков DSL, она может быть трансформирована к другой модели с более низким уровнем DSL. Это позволяет интегрировать между собой разные части процессов жизненного цикла, написанные на разных языках DSL.

3.3. Описание модели характеристик процессов жизненного цикла

Для проведения доменного анализа стандарта жизненного цикла используется FODA (Feature-Oriented Domain Analysis). Отличительной особенностью представления процессов ЖЦ является диаграмма зависимостей между характеристиками. Нотация диаграмм характеристик процессов выполняется языком FDL (Feature Definition Language), позволяющим описывать:

- 1) атомарные и композиционные характеристики, имена которых определяются и используются в разных процессах;
- 2) необязательные (optional) и обязательные характеристики (mandatory) выражений, которые относятся к замкнутым конструкциям all();
- 3) альтернативные характеристики (exclusive – выбор) через one-of();
- 4) характеристики по умолчанию (default).

Результат трансляции характеристик в FDL может быть выполнен в XML для обмена информацией (XML Metadata Information Exchange format) и может быть импортирован в систему моделирования UML при генерации классов. Подход к описанию модели характеристик предметной области использован при разработке вариантов жизненного цикла программной системы и конфигурировании разных процессов с помощью данной модели генерируются необходимые варианты жизненного цикла для реализации определенного класса программных систем.

3.4. Описание процессов жизненного цикла средствами DSL, Protégé

Для описания онтологии домена жизненного цикла взят язык Eclipse DSL. В этом языке имеются достаточные средства разработки графических моделей жизненного цикла. Описание основных процессов домена жизненного цикла с помощью инструментария DSL Tools VS.Net. На рис. 6 приведена схема онтологии основных процессов на языке DSL.

Типы отношений в данном графическом представлении задают основную логику процессов домена жизненного цикла. В каждом классе заданы методы и поля, необходимые для функционирования [22].

Процессы поддержки включают в себе все процессы, которые выполняются после построения системы и поддержки его работоспособности. Их онтологическая структура отражает структуру основных процессов жизненного цикла. Затем осуществляется генерация имеющихся графических

моделей в текстовое представление на языке XML. При этом процесс тестирования жизненного цикла рассмотрен отдельно с помощью инструментального средства Protégé. В нем он представляется соответствующими правилами и отношениями.

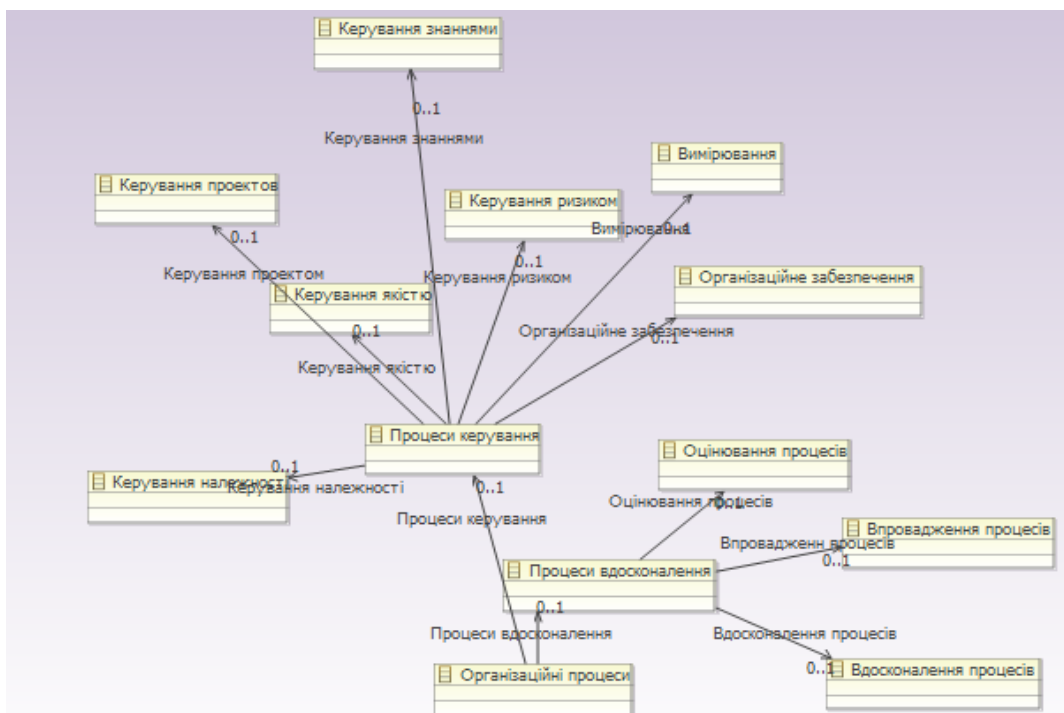


Рис. 6. Онтология основных процессов жизненного цикла

Текстовое описание процессов жизненного цикла задается на языке XML. Ошибки в графическом описании были найдены дизайнером и исправлены с помощью соответствующего редактора. Результат каждого процесса дается на XML.

Далее приведен пример фрагмента описания основных процессов жизненного цикла на языке XML:

```
<?xml version="1.0" encoding="utf-8"?>
<AssociationLine Name = "Определение требований"
  Type="Main. Определение требований"
  ManuallyRouted = "true" FixedFromPoint = "true"
</AssociationLine>
<AssociationLine Name = "Интеграция_ПС" Type =
"Main.Интеграция_ПС" />
<AssociationLine Name = "Инсталляция" Type = "Main.Инсталляция" />
<AssociationLine Name = "Эксплуатация" Type = "Main.Эксплуатация" />
<Property Name = "Интеграция_ПС" />
<Property Name = "Инсталляция" />
<Property Name = "Анализ требований" />
<Property Name = "Эксплуатация" />...
```


Это описание преобразуется в реализационный машинный код. В рамках общего процесса жизненного цикла разработана онтология процесса тестирования средствами Protégé [24, 25]. Элементы онтологии представлены на сайте ИТК (<http://7dragons.ru/ru>) в разделе «Онтология». На этом сайте можно сделать новую онтологию, используя систему Protégé и DSL Tool VS.Net.

Заключение

Описаны методы и средства онтологии для использования в глобальной семантической сети, пригодные для работы с разными предметными областями. Приведено описание и реализация конкретных онтологий для научной математической области – «Вычислительная геометрия» и домена «Жизненный цикл стандарта ISO/IEC 12207-2007». Дано описание основных задач этих доменов и приведены онтологические схемы. Подходы к описанию онтологии обсуждались на семинарах и конференциях (ICTERI-2013-2014 и Science ant Information-2015), а также описаны в статьях и размещены на сайте (<http://7dragons.ru/ru>) с целью использования при преподавании курса «Программная инженерия» в МФТИ. Вычислительная геометрия реализована на сайте КНУ <http://cg.unicyb.kiev.ua>, а также Computational Geometry Pages.

Работа поддержана грантами Российского фонда фундаментальных исследований № 15-07-02355_а, № 15-01-02362_а и № 16-01-00352_а.

Литература

1. Semantic Web programming. S. Hebel, M. Fisher, R. Blace, A. Peter-Lopes, Willey Publishing Inc., 2008, <http://semwebprogramming.org>
2. OWL Web Ontology Language. W3C Working Draft 29 July 2002. Latest version is available at <http://www.w3.org/>
3. Semantic Web, Representation of data on the World Wide Web, based on the RDF standards, <http://www.w3.org/2001/sw/>
4. OASIS/ebXML Registry Information Model, <http://www.oasis-open.org/cojmmetees/specs/ebrim>
5. D. Roman, H. Lausen, U. Keller (eds.): Web Service Modeling Ontology (WSMO: <http://www.w3.org/Submission/wsmo>)
6. www.swebok.org
7. Web Service Glossary, <http://www.w3.org/TR/ws-gloss>
8. Mens., Van Gorp P, Czarnecki K. A. Taxonomy of Model Transformation.– <http://drops.dagstuhl.de/2—5/11>
9. D. Roman, H. Lausen, U. Keller (eds.): Web Service Modeling Ontology (WSMO: <http://www.w3.org/Submission/wsmo>)

10. J. de Bruijn, H. Lausen, R. Krummenacher, A. Polleres, L. Predoiu, M. Kifer, and D. Fensel: The Web Service Modeling Language (WSML). <http://www.w3.org/Submission/wsml>
11. Semantic Annotations for WSDL and XML Schema. W3C Recommendation. <http://www.w3.org/TR/sawSDL/>
12. Web Service Semantics – WSDL–S.W3C Member Submission. <http://www.w3.org/Submission/WSDL–S/>
13. Web Service Glossary // <http://www.w3.org/TR/ws–gloss>.
14. Франко Р., Препарата Ф., Шеймос М. Вычислительная геометрия. Введение. — М. Мир, 1989. — 478 с. — ISBN 5-03-001041-6.
15. A.R. Forrest, «Computational geometry», Proc. Royal Society London, 321, series 4, 187–195 (1971)
16. Анісимов А.В., Терещенко М.В. Навчально-методичний посібник з курсу "Обчислювальна геометрія та комп'ютерна графіка" факультету Кібернетики КНУ імені Тараса Шевченка.
17. Computational Geometry
18. Computational Geometry Pages, Geometry In Action
19. "Strategic Directions in Computational Geometry—Working Group Report" (1996), Journal of Computational Geometry.
20. Standard ISO/IEC 12207-2007 Life Cycle
21. Lavrischeva Ekaterina. Ontological Approach to the Formal Specification of the Standard Life Cycle, "Science and Information Conference-2015", July 28-30, London, UK, www.conference.thesai.org. - p.965-972.
22. Лаврищева Е.М. Software Engineering компьютерных систем. Парадигмы, Технологии, CASE – средства. – Наук. Думка, 2014. - 282 с.
23. Лаврищева Е. М., Карпов Л. Е., Томилин А. Н. Системная поддержка бизнес задач в глобальной информационной сети. -Труды конференции «Научный сервис в сети Интернет -2015», 21-26 сентября 2015, Новороссийск, с. 193-218.
24. Лаврищева К. М. Подход к формальному представлению онтологии ЖЦ программных систем Вісник КГУ, серія фіз.-мат.наук. – 2013. – №4. – С. 140–149.
25. Лаврищева К. М. Отологічне подання життєвого циклу ПС для загальної лінії виробництва програмних продуктів.- Теоретичні та прикладні аспекти побудови програмних систем.-ТАAPSD-2013.- 25.05-02.06.2013, Ялта.-с.81-90.