

Общероссийский математический портал

Д. В. Кудрявцев, А. М. Беглер, Т. А. Гаврилова, И. А. Лещева, М. В. Кубельский, О. Н. Тушканова, Метод коллективной визуальной разработки онтологического графа знаний, *Искусственный интеллект и принятие решений*, 2019, выпуск 1, 27–38

DOI: 10.14357/20718594190103

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением
<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 3.17.76.163

26 декабря 2024 г., 10:20:38



Метод коллективной визуальной разработки онтологического графа знаний*

Д.В. Кудрявцев¹, А.М. Беглер¹, Т.А. Гаврилова¹, И.А. Лещева¹, М.В. Кубельский¹, О.Н. Тушканова^{II,III}

¹ Институт «Высшая школа менеджмента» СПбГУ, г. Санкт-Петербург, Россия

^{II} Санкт-Петербургский институт информатики и автоматизации РАН, г. Санкт-Петербург, Россия

^{III} ООО "Информационные крылья", Санкт-Петербург, Россия.

Аннотация. В статье предложен метод создания графа знаний с акцентом на визуализации и групповой работе, который был создан в рамках проекта с автомобилестроительной компанией. Метод ориентирован на создание графа знаний со сложной структурой свойств, связанных с одним классом и его подклассами, и в котором число онтологических классов невелико. Особенностью предлагаемого метода также является необходимость активного привлечения экспертов к созданию графа знаний, поскольку они и их коллеги являются будущими пользователями (внутренними клиентами) системы управления знаниями. С помощью предлагаемого метода был создан пилотный граф знаний, охватывающий более 50 сборочных единиц и около 400 типов свойств.

Ключевые слова: граф знаний, онтология, метод разработки графа знаний, инструменты визуализации, коллективная разработка.

DOI 10.14357/20718594190103

Введение

Граф знаний представляет собой набор объектов (с атрибутами), которые связаны друг с другом типизированными отношениями. Типы сущностей и отношений определены в схемах, которые называются онтологиями [1]. Онтология – это формальная, явная спецификация общей концептуализации [2], состоящая из иерархии понятий предметной области, связей между ними и ограничений, которые действуют в рамках этой модели. Связи могут быть разных типов (например, «есть», «состоит из», «является исполнителем» и т. д.). Термин «онтология» пересекается с термином «граф знаний», поскольку, при наполнении онтологии экземплярами, происходит ее превращение в граф знаний (или базу знаний). Однако основным фокусом онтологии остаются классы,

свойства и аксиомы (TBox), в то время как в графе знаний больше внимания уделяется экземплярам (ABox).

Первые методологии по разработке онтологий были предложены в начале 90-х годов. Одной из первых появилась статья, описывающая процесс создания онтологии верхнего уровня Суэ [3], несколькими годами позже – методологии по результатам работы над Enterprise Ontology [4] и онтологией Toronto Virtual Enterprise (TOVE) [5], сформировавшие первое поколение методологий разработки онтологий. В тот же период стали появляться методологии для совместной разработки, основная цель которых – достижение консенсуса между локальными группами, принимающими участие в разработке [6]. Такой подход применяется в большинстве методологий третьего поколения [7], которые фокусируются на совместной разработке онтологий распределенными команда-

* Работа выполнена при финансовой поддержке РФФИ (проект № 17-07-00228).

Беглер Алёна Маратовна. E-mail: alena.begler@gmail.com

ми. Ряд онтологий предыдущего, второго, поколения, таких как Methontology [8] и On-To-Knowledge [9] также подразумевают совместное создание онтологий командой, состоящей из участников с разными ролями (инженер по знаниям, эксперт предметной области). Такие подходы фокусируются на обеспечении непрерывного взаимодействия между специалистами. Второе поколение методологий, таким образом, рассматривает совместную централизованную разработку онтологии, тогда как третье – децентрализованную.

Появление методов децентрализованной разработки не означает отказа от развития методов предыдущего поколения. Наоборот, отмечается необходимость быстрой разработки прикладных онтологий для нужд производственного процесса [10, 11]. При этом, задачи, решаемые в прикладных областях, ориентированы не столько на создание онтологии, сколько на создание баз знаний на ее основе. Таким образом, возрастает важность наполнения онтологии. Это объясняет актуальность создания метода для групповой разработки графа знаний. Требования к создаваемому методу были специфицированы в соответствии с требованиями к графу знаний и парадигмой проектирующего исследования (design science research) [12]:

- граф знаний, являющийся финальным результатом применения метода, имеет сложную структуру свойств, которые связаны с одним классом и его подклассами;
- ABox (описание экземпляров) и TBox (структура свойств/классов) одинаково важны;
- главный источник информации для создания графа знаний – слабоструктурированные данные о предметной области;
- необходимость включения специалистов предметной области и пользователей на всех этапах разработки.

Целью данной статьи является предложение и описание метода создания графа знаний, отвечающего перечисленным требованиям. В работе рассмотрено решение описанных задач с применением существующих методологий и предложен ряд новых дополнений, апробированных при разработке онтологического графа знаний для автомобилестроительной компании.

В Разделе 1 данной работы представлен обзор существующих методологий разработки онтологий и графов знаний. Раздел 2 посвящен краткому введению в предлагаемые к ис-

пользованию инструменты визуальной разработки. В Разделе 3 изложен метод разработки онтологического графа знаний и его демонстрация для описания сборочных единиц автомобилестроительного предприятия.

1. Обзор методологий разработки онтологий и графов знаний

Существующие методологии сконцентрированы на разработке схемы онтологии и уделяют меньше внимания наполнению онтологии (что важно для графа знаний). Тем не менее, эти методологии составляют ядро разработки графа знаний. В Табл. 1 универсальные методологии централизованной разработки онтологий и графов знаний «с чистого листа», опубликованные в научной литературе, рассмотрены в соответствии с инструментами, используемыми на разных этапах разработки (если инструмент не указан в соответствующих публикациях, в ячейке таблицы стоит прочерк). Этапы разработки соответствуют методологии NeOn [13], вобравшей в себя лучшие практики предшествующих подходов: спецификация требований к онтологии, сбор знаний (формирование глоссария), концептуализация (создание неформальной модели), формализация (создание полужформальной модели), программная реализация (на формальном языке). В последовательность этапов, рассматриваемых в обзоре онтологий, не включен этап планирования, как не относящийся непосредственно к разработке онтологии.

В обзор включены: методология Ушольда и Грюнингера [14], Methontology [8], Ontology Development 101 [15], On-To-Knowledge [9], методология основанная на методе Делфи [16], методология с использованием медиации [17], UPON [18, 19], DILIGENT [20], HCOME [21], Rapid-OWL [22], ontology maturing [23], NeOn [13] и методология построения графа знаний [1].

Из Табл. 1 видно, что для этапов концептуализации, формализации и программной реализации (включающей, в случае графа знаний, еще и наполнение) конкретные инструменты зачастую не рассматриваются. Так, на этапе формализации предлагается только формирование типов отношений и интервью с экспертами (методология On-To-Knowledge). На этапе программной реализации – использование паттернов дизайна онтологий (NeOn Methodology) и алгоритм создания экземпляров класса

Табл. 1. Инструменты, предлагаемые существующими методологиями для реализации разных этапов разработки онтологии

Методология	Этап разработки				
	Спец. требований	Сбор знаний	Концептуализация	Формализация	Программная реализация
Ушольд и Грюнингер	Вопросы для проверки компетентности	Мозговой штурм	Моделирование «из середины»	-	-
Methontology	Вопросы о цели разработки онтологии	Мозговой штурм, формальный и неформальный анализ текстов	Моделирование «из середины», разделение терминов на глаголы и существительные, таблицы для разных типов сущностей	-	Изучение сравнительных характеристик языков
Ontology Development 101	Вопросы для проверки компетентности и о цели разработки	-	Список терминов с разделением на классы и свойства	-	Выбор класса – создание экземпляра – присвоение значения
On-To-Knowledge	-	-	Граф (узлы и ребра, которые могут быть дополнены описаниями)	Формирование типов отношений, интервью с экспертами	-
Delphi-based	Проектные требования, определение содержания и стандартов оценки	Опрос экспертов методом Дельфи	-	-	-
Knowledge mediation	-	Мозговой штурм, глоссарий	-	-	-
UPON	Достижение консенсуса, вопросы для проверки компетентности, интервью	Мозговой штурм, анализ документов и примеров использования	Категоризация концептов	-	Рекомендованный формализм: OWL
DILIGENT	-	Моделирование «из середины»	-	-	-
HCOME	Вопросы для проверки компетентности	Моделирование «снизу вверх»	-	-	Дескриптивная логика NeoClassic
RapidOWL	-	Моделирование «из середины», общение с экспертами	-	-	Рекомендованный формализм: RDF
Ontology maturing	-	Моделирование «снизу вверх» – получение тегов от сообщества	Анализ сходств и различий тегов	-	-
NeOn	Вопросы для проверки компетентности	-	-	-	Использование паттернов дизайна онтологий
Construction of Enterprise Knowledge Graphs	Вопросы для проверки компетентности	Идентификация и анализ источников данных	-	-	Трансформация идентифицированных данных в RDF

(Ontology Development 101). Остальные методологии предлагают формализм для программной реализации онтологии, но не инструменты для трансформации полуформальной модели в формальную.

2. Инструменты визуальной разработки онтологического графа знаний

Во время разработки онтологии были использованы два инструмента визуализации для

общения внутри команды инженеров по знаниям или с экспертами и перспективными пользователями: интеллект-карты [24] и таблицы. Эти форматы были выбраны как наиболее легко воспринимаемые экспертами предметной области и продемонстрировавшие эффективность для поддержки разработки онтологии.

Интеллект-карты хороши для ранних стадий онтологического инжиниринга, так как они подходят для представления слабоструктурированных областей [25, 26]. Кроме того, они упрощают процесс коммуникации, так как могут быть легко восприняты и отредактированы экспертами предметной области. На следующем шаге предлагаемого метода узлы интеллект-карт были превращены в свойства. При этом каждый новый уровень интеллект-карты задавал отношение «родитель-потомок» между соответствующими свойствами. Аналогичный прием используется в подходах *OntoEdit* [27] и *Křemen* с соавторами [28], в которых концепты интеллект-карты объединялись отношениями *subConceptOf* и *subClassOf* соответственно.

Таблицы часто применяются для автоматического и полуавтоматического создания онтологий [29]. Однако они могут быть использованы и для ручной разработки, в частности, для решения следующих задач: создание словаря (в этом случае таблицы упрощают поиск синонимов и связей между понятиями [30]), обогащение онтологии и идентификация структуры классов [31]. Дополнительное преимущество таблиц заключается в том, что они позволяют автоматизировать процесс наполнения онтологии.

3. Описание и применение метода визуальной коллективной разработки онтологического графа знаний VITON

Предлагаемый метод визуальной коллективной разработки онтологического графа знаний, соответствующего требованиям, представленным во введении, получил обозначение VITON (*V*isual *col*lective *de*velopment of the *ON*tological *kn*owledge *g*raph).

На верхнем уровне VITON следует логике существующих методологий онтологического инжиниринга, а именно, методологии *NeOn* [13]. Однако метод обладает некоторыми дополнительными особенностями, которые поз-

воляют удовлетворить приведенные выше требования:

- фокус на свойствах и их структуре;
- использование интеллект-карт (высокоуровневые структуры свойств, описания экземпляров) для взаимодействия с экспертами и будущими пользователями (внутренними потребителями) системы;
- использование таблиц (матриц «Свойства – Экземпляры») для структурирования свойств, анализ их использования в экземплярах и поддержания взаимодействия с командой заказчика;
- объединение разработки онтологии с описанием экземпляров, где каждое описание экземпляра дает новые свойства для онтологии.

Процесс построения графа знаний следовал Сценарию 2 методологии *NeOn* [13]. Этот сценарий включает в себя повторное использование не онтологических ресурсов в дополнение к основным стадиям: спецификация требований, поиск потенциальных ресурсов знаний для повторного использования, планирование и разработка. Разработка, в свою очередь, разделяется на этапы концептуализации, формализации и реализации на формальном языке.

Параллельно с процессом разработки графа знаний решались две поддерживающие задачи: сбор знаний и документирование. Сбор знаний был реализован во время анализа экземпляров и включал анализ стандартов, технических спецификаций и интервью с экспертами. Документирование включало в себя: контроль версий интеллект-карт, таблиц и онтологий (файлы контроля версий); фиксацию принятых решений (заметки по итогам встреч, документ для отслеживания принимаемых решений); библиотеку используемых источников (список ресурсов знаний, *wiki*-страницы со ссылками на связанную с экземплярами информацию).

Метод VITON был применен для формирования пилотного графа знаний для сборочных единиц электроавтомобилей, включающего примерно 50 экземпляров сборочных единиц и 400 свойств. После первоначального формирования графа, предполагается его дальнейшая доработка и заполнение сотрудниками предприятия.

Задачи коллективного создания онтологического графа знаний и разработки метода возникли в связи с промышленной необходимостью разработать такой граф для автомобилестроительной компании. Элементы графа знаний представляют собой сборочные единицы (СЕ) различ-

ных моделей автомобилей, в основном электрогрузовиков, производимых компанией. Созданный граф знаний служит основой для новой корпоративной системы управления знаниями, в которой пользователи (инженер, менеджер по закупкам, экономист и т. д.) смогли бы найти СЕ или ее аналоги, в которых они нуждаются, а также всю соответствующую информацию.

Создание подобной системы имеет принципиальное значение для компании, поскольку выпускаемые автомобили преимущественно собираются из СЕ, доступных на рынке. Поиск информации о таких СЕ является одной из наиболее затратных по времени задач, решаемой инженерами в процессе проектирования. Предполагается, что создаваемая система будет интегрировать как информацию о СЕ, уже используемых в компании, так и об аналогичных СЕ, которые представлены на рынке в каталогах поставщиков. В будущем граф знаний может использоваться в роботизированном производстве, которое также развивается в компании. Заказчиком метода VITON и графа знаний является ИТ-отдел, который выполняет оцифровку всех бизнес-процессов предприятия.

3.1. Сбор требований и планирование разработки

Требования к графу знаний были согласованы с пользователем графа знаний (автопроизводителем):

- требуемый граф знаний должен был ориентирован на структуру свойств: малая часть свойств является общей для обобщенного класса включающего все СЕ), а большая часть свойств зависит от типа СЕ;
- ABox (описание СЕ/экземпляров) и TBox (структура свойств) одинаково важны;
- высокое качество графа знаний для использования в качестве «золотого стандарта данных» и демонстрации содержания.

Время на проект было ограничено заказчиком до трех месяцев. Таким образом, на стадии планирования проекта определялось его содержание (состав работ) и происходило формирование проектной команды. Проектная команда включала в себя четыре группы специалистов:

- аналитики (2 человека), осуществлявшие поиск источников знаний, создание интеллектуальных карт СЕ и трансформация интеллектуальных карт в таблицы;

- онтологи (2 человека), создававшие макет онтологии и непосредственно граф знаний, конвертируя интеллектуальные карты в RDF с помощью редакторов онтологий;

- эксперты предметной области (2 человека), которые были задействованы в процессе каждый раз, когда новая порция СЕ была описана, для проверки точности описания, и после создания нового макета онтологии для его утверждения;

- технический писатель-методолог (1 человек), консультирующий команду по использованию методов и описывающий внутренние процедуры.

3.2. Повторное использование потенциальных ресурсов знаний

В соответствии с методологией NeOn, повторное использование потенциальных ресурсов знаний включало ряд процедур:

- поиск работ, описывающих онтологии продукта/изделия и автопроизводства, промышленных стандартов и интернет ресурсов (онлайн магазины, сайты поставщиков);
- оценка источников ресурсов (приоритет отдавался известным разработчикам, таким как ISO и NIST, и мнениям экспертов предметной области);
- выбор ресурсов для повторного использования в соответствии с критериями оценивания.

Идентифицированные ресурсы знаний были использованы на следующих этапах:

- базовая модель продукта Core Product Model (CPM) [32], описание продукта в schema.org [33], основанное на онтологии GoodRelations [34], а также области информации о продукте; идентифицированные в системах управления жизненным циклом продукта [35] были использованы как основа для разработки макета онтологии в форме интеллектуальной карты и формирования словаря на этапе концептуализации;

- стандарты ISO 14050:2009(en), ISO 10303-1:1994(en), ISO/IEC 15288:2008 и STEP AP 242 использовались для уточнения словаря и определения понятий;

- стандарт IEC 81346 «Производственные системы, установки и оборудование и промышленная продукция. Принципы структурирования и условные обозначения» использовался для классификации объектов по целевому

назначению на стадиях формализации и программной реализации;

- список единиц измерения элементов «Codes for Units of Measurement used in the International Trade» [36] использовался как часть онтологии для описания экземпляров на стадиях формализации и программной реализации.

3.3. Концептуализация (применение интеллект-карт)

На этой стадии были использованы интеллект-карты (как язык описания) и две техники моделирования.

Сначала было применено нисходящее моделирование и создан макет онтологии в форме интеллект-карты. Он позволил определить главные свойства центрального класса («Артефакт», суперкласс CE) и стал главным инструментом для описания экземпляров (конкретных CE). Первоначальная структура была предложена заказчиком и скорректирована в соответствии с Core Product Model и описанием продукта в schema.org. Визуальный обобщенный структурный макет онтологии был создан в форме интеллект-карты группой аналитиков в течение нескольких итераций и обсуждений с заказчиком. Рис. 1 показывает текущую версию макета. Она не содержит формализованных отношений между понятиями (ребра не подписаны), так как отображает основные аспекты ключевого понятия («Артефакт»).

Затем было применено восходящее моделирование. Были созданы интеллект-карты для набора пилотных CE, предоставленных заказчиком в форме таблиц со списком CE со ссылками на сайты производителей (около 50 CE, которые затем стали экземплярами онтологии). Эти описания были созданы на основе макета и затем оценены группой экспертов. Рис. 2 показывает фрагмент заполненного макета онтологии для CE (для воздушного компрессора серии ERC). В этой интеллект-карте по сравнению с макетом онтологии:

- появился новый элемент высокого уровня (помечен серым цветом): Has connection («имеет связь»). Этот элемент затем будет добавлен в макет;
- уточнены свойства более низких уровней, например, свойство Output («выход») было дополнено тремя дочерними свойствами: Noise («шум»), Free air delivery («свободное нагнетание воздуха»), Rated pressure («разрешенное давление»);
- были добавлены значения свойств. Например, для свойства Noise («шум») известно минимальное и максимальное значения в децибелах.

Эти два процесса реализовывались итеративно: интеллект-карты для каждой CE основывались на макете и сам макет непрерывно изменялся в соответствии с новыми свойствами, идентифицированными во время анализа CE. Также во время этой работы был собран словарь понятий с определениями.

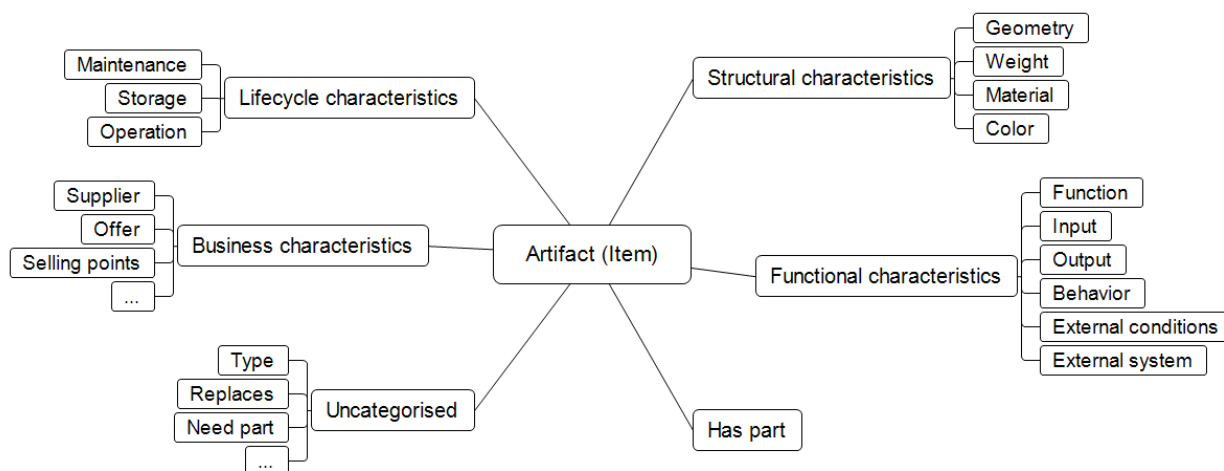


Рис. 1. Текущая версия макета онтологии

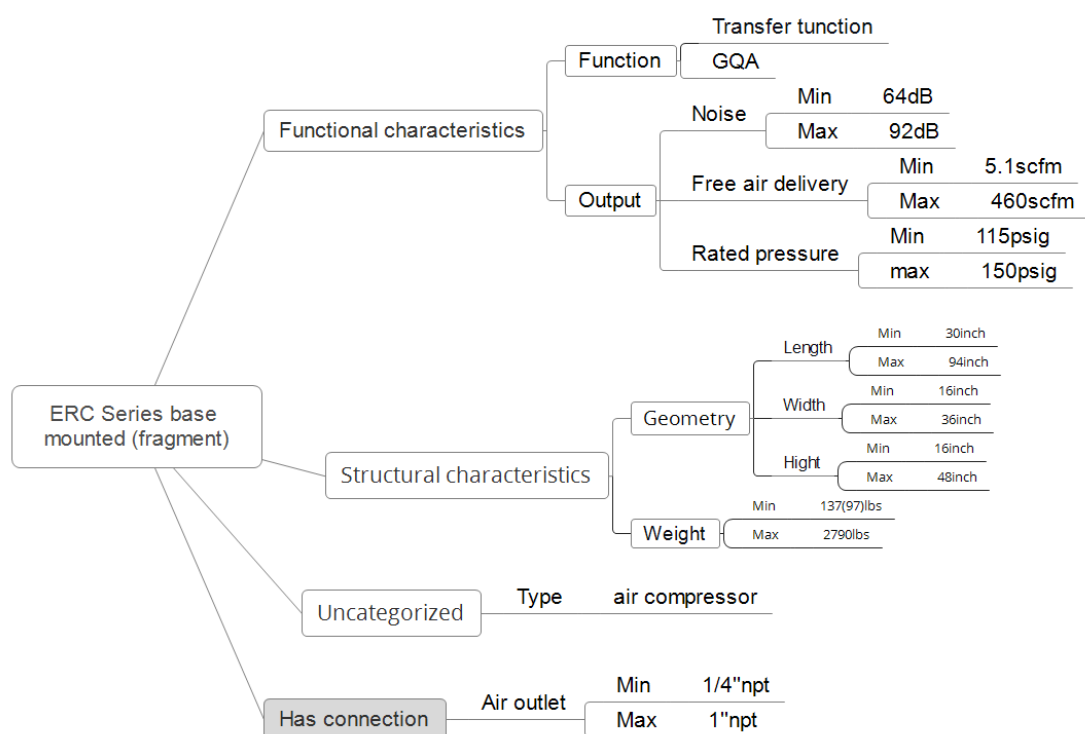


Рис. 2. Фрагмент заполненного макета для СЕ воздушный компрессор серии ERC. Серый фон – для нового свойства артефакта, которое должно быть добавлено в макет

3.4. Формализация (применение таблиц)

После того, как были созданы интеллект-карты для примерно 50 СЕ и зафиксирован макет онтологии (т.е. дополнительный анализ СЕ немного привносил в макет), была выполнена формализация свойств с использованием таблиц. Таблица со свойствами служила промежуточным этапом между интеллект-картами и онтологией. Главной целью создания таблицы было формирование иерархии свойств и словаря. Интеллект-карты для разных СЕ содержали характеристики, которые являются дочерними свойствами того же свойства высокого уровня. Например, воздушный компрессор серии ERC имеет выходную характеристику Free air delivery («свободное нагнетание воздуха»), которая связана со свойством Output («выход»), так же как свойство Air flow («поток воздуха») другого экземпляра. Кроме того, одни и те же характеристики СЕ могут быть названы по-разному в разных каталогах поставщиков, поэтому внесение в таблицу помогало выявлять синонимы. Похожее использование таблиц для создания онтологии было упомянуто в подходах Rubin'a [30] и Ontorat [31].

Преобразованные интеллект-карты составили основу таблицы (Табл. 2 представляет фрагмент таблицы для СЕ на примере воздушного компрессора серии ERC). Техника преобразования, схожая с ранее описанными техниками для трансформации интеллект-карт [27, 28], была применена с некоторыми дополнениями:

- СЕ сформировали столбцы в таблице (субъекты).
- Понятия из макетов онтологий были преобразованы в свойства высокого уровня, а элементы низкого уровня из интеллект-карты СЕ – в дочерние свойства. В таблице они сформировали строки (предикаты). Во время преобразования имена свойств были унифицированы в соответствии с выбранной формой именования, были добавлены человекочитаемые имена в качестве ярлыков, а также описание.
- Самый нижний уровень интеллект-карты был помещен в ячейки таблицы (объекты). Для каждой СЕ все связанные свойства добавлялись в таблицу вместе с их размерностями (при наличии). На пересечении строки (свойства) и столбца (СЕ) могла быть либо пустая клетка (в случае отсутствия свойства у данного экземпляра), либо символ «+» или единицы измерения в случае

Табл. 2. Фрагмент таблицы свойств для CE воздушный компрессор серии ERC

Серая заливка означает свойства высокого уровня, у которых нет значений (только дочерние свойства)

Свойство	Ярлык	Описание	CE	
			Серия ERC	...
hasFunctionalCharacteristics	Функциональные характеристики	«Атрибут или характеристика в использовании и результативности продукта» (Источник: ISO 14050:2009(en), 8.3.3.2)		
hasFunctionCharacteristics	Функция	Предположительное поведение артефакта		
hasType	Тип артефакта	Неформальное описание типа (мотор, батарея, и т.д)	+	
hasFunctionType	Тип функции	Тип функции в соответствии с IEC 81346	+	
hasOutputCharacteristics	Выход	Весь выход, который может быть предоставлен артефактом в случае правильного функционирования		
hasOutputAirCharacteristics	Характеристики выхода воздуха	Группа характеристик для описания выхода воздуха		
hasOutputFreeAirDelivery	Свободная подача воздуха			scfm
hasRatedPressure	Разрешенное давление			psig
...				

наличие свойства у экземпляра (единицы измерения указывались для свойств, которые имеют конкретные численные значения, измеренные в указанных единицах). Например, если CE имело свойство Size («размер»), измеренное в миллиметрах, имя свойства добавлялось в колонку «Свойства» под соответствующей группой или подгруппой свойств, а код «мм» добавлялся в ячейку на пересечении свойства с данной CE. В случае если единица измерения отсутствовала (например, качественное свойство), в ячейку вносился знак «+».

Создание таблицы состояло из трех этапов:

1. Первоначальный ввод свойств в соответствии с интеллект-картой макета онтологии.

2. Уточнение общей структуры свойств с учетом их семантической близости и уровня обобщения.

3. Ввод всех CE в созданную структуру свойств (с указанием имеющихся свойств и, при необходимости, их значений), утверждение и финализация структуры, добавление ярлыков и описаний свойств.

3.5. Программная реализация и наполнение (применение редактора онтологий)

Иерархия свойств, созданная на предыдущем этапе, была затем преобразована в описание на языке OWL с помощью редактора онтологий (WEB PROTÉGÉ). Переход состоял из трех шагов.

На первом шаге свойства были импортированы в редактор онтологий. Во время их переноса общая иерархия свойств была разделена на две отдельные иерархии для свойств-объектов (object property) и свойств-значений (datatype property). При этом свойства верхнего уровня составляли верхний уровень классификации для обоих типов свойств (например, свойство с именем hasFunctionalCharacteristics существовало и в иерархии свойств-объектов, и в иерархии свойств-значений).

Затем была создана иерархия классов с четырьмя классами верхнего уровня:

- Artifact («артефакт») – для экземпляров сборочных единиц и их частей;

- Organization («организация») – для экземпляров организаций поставщиков и производителей;

- QuantitativeValue («количественное значение») – для экземпляров количественных значений свойств;

- Reference («словари») – для экземпляров значения свойств импортированных из внешних источников (например, в этот класс входил список единиц измерения).

На финальном шаге все значения были импортированы или как значения для свойств-значений (например, уникальные качественные значения, как неформальные описания), или как экземпляры, связанные с артефактом (конкретной CE) свойством-объектом. Рис. 3 представляет фрагмент онтологического описания для воздушного компрессора серии ERC.

Создание экземпляров состояло из нескольких шагов:

1. Артефакты (CE) были импортированы как экземпляры класса Artifact со свойствами

rdfs:label (человекочитаемое название) и rdfs:description (описание, если доступно).

2. Организации импортировались как экземпляры класса Organization со свойствами rdfs:label (человекочитаемое название) и hasURL (ссылка на сайт организации).

3. Значения свойств из других словарей импортировались как экземпляры класса Reference.

4. Количественные значения импортировались как экземпляры класса QuantitativeValue. Каждый экземпляр описывал значение или диапазон значений. Например, компрессор серии ERC имеет свойство hasFreeAirDelivery («свободное нагнетание воздуха») с диапазоном значений от 5.1 до 460 scfm. Это импортируется как экземпляр «5.1-460scfm» класса Flow («поток», подкласс QuantitativeValue) со свойствами-значениями hasMaxValue (максимальное значение: 460), hasMinValue (минимальное значение 5.1; hasUnitCode, код единицы измерения: scfm).

The screenshot displays two panels from the WEB PROTÉGÉ ontology editor. The left panel, titled 'Class Hierarchy', shows a tree structure of classes. The right panel, titled 'Individual: ERC Series Base Mounted', shows the details for a specific instance.

Class Hierarchy:

- owl:Thing
 - Artifact
 - Assembly
 - Component
 - Organization
 - QuantitativeValue
 - Current
 - Flow
 - Frequency
 - Phase
 - Power
 - Pressure
 - Proportion
 - Size
 - SoundNoise
 - Speed
 - Temperature
 - Time
 - Voltage
 - Volume
 - Weight
 - Reference
 - Colour
 - FunctionType

Individual: ERC Series Base Mounted

IRI
http://webprotege.stanford.edu/RwkSbKewLm27vXoIdw0VEQ

Annotations

rdfs:label	ERC Series Base Mounted
Enter property	Enter value

Types

Assembly

Enter a class name

Relationships

hasAirCoolingSystem	CoolingSystem_ERCSeries
hasAirOutlet	Size1/4-1NPT
hasAirReceiverTank	Volume90gallon
hasConfigurations	ERC Series Stacked System
hasDocument	http://cdn2.hubspot.net/hub/37
hasDocument	http://www.matteicomp.com/ai
hasFunctionType	GQA
hasIdleSpeed	Speed1800rpm
hasInputFreeAirDelivery	5.1-460scfm
hasInputFrequency	Frequency60Hz

Рис. 3. Фрагмент онтологического описания CE компрессор серии ERC в программе WEB PROTÉGÉ

Заключение

В работе описан метод создания онтологического графа знаний предприятия. Метод был протестирован на сборочных единицах электроавтомобилей, однако он может быть широко использован и в других областях, где граф знаний имеет сложную структуру свойств, ассоциированную с одним классом и его подклассами, и количество классов небольшое.

Потребность в создании нового метода VITON обусловлена тем, что в имеющихся методологиях разработки онтологий недостаточно освещены конкретные инструменты, использующиеся на различных этапах, в особенности это касается этапов формализации и программной реализации. В данной работе подробно описано применение инструментов визуальной разработки (на этапах концептуализации и формализации), а также наполнения онтологии (на этапах формализации и программной реализации).

Выбор инструментов визуальной разработки (интеллект-карт и таблиц) связан с тем, что во время разработки графа знаний предприятия часто необходимы групповая разработка и обсуждение. В этом случае визуализация графа позволяет членам команды разработки быстрее и эффективнее понимать друг друга. Зачастую, одной из главных проблем, возникающих в процессе разработки графов знаний предприятия, является недостаток или неоднозначность информации, разное понимание терминологии и приоритетности сборочных единиц у экспертов предметной области. Визуальное описание фрагментов графа знаний в виде интеллект-карт и таблиц помогает сфокусировать внимание экспертов в ходе обсуждений для уточнения спорных моментов.

Предложенный метод был апробирован при разработке пилотного графа знаний, включающего более 50 сборочных единиц и 400 типов свойств. Созданный граф используется ИТ-отделом компании-заказчика в качестве «каркаса» информационной системы для оцифровки всех бизнес-процессов предприятия.

Литература

- Villazon-Terrazas B. et al. Construction of Enterprise Knowledge Graphs (I) // *Exploiting Linked Data and Knowledge Graphs in Large Organisations*. Springer, 2017. P. 87–116.
- Gruber T.R. Toward principles for the design of ontologies used for knowledge sharing // *Int. J. Hum. Comput. Stud.* 1995. Vol. 43. P. 907–928.
- Lenat D.B. et al. Cyc: Toward Programs With Common Sense // *Commun. ACM*. 1990. Vol. 33, № 8. P. 30–49.
- Uschold M., King M. Towards a Methodology for Building Ontologies. 1995. 13 p.
- Gruninger M., Fox M.S. The Role of Competency Questions in Enterprise Engineering // *Benchmarking — Theory and Practice*. 1995. P. 22–31.
- Skuce D. Conventions for reaching agreement on shared ontologies // *Proceedings of the 9th Knowledge Acquisition for Knowledge Based Systems Workshop*. 1995.
- Simperl E., Luczak-Rösch M. Collaborative ontology engineering: A survey // *Knowl. Eng. Rev.* 2014. Vol. 29, № 1. P. 101–131.
- Fernández-López M., Gómez-Pérez A., Juristo N. METHONTOLOGY: From Ontological Art Towards Ontological Engineering // *AAAI-97 Spring Symposium Series*. 1997. Vol. SS-97-06. 33–40 p.
- Sure Y., Staab S., Struder R. On-To-Knowledge Methodology // *Handbook on Ontologies*. 2004. P. 117–132.
- Ohgren A., Sandkuhl K. Towards a Methodology for Ontology Development in Small and Medium-Sized Enterprises // *IADIS International Conference on Applied Computing*. 2005. P. 369–376.
- Слободюк А.А., Маторин С.И., Четвериков С.Н. О подходе к созданию онтологий на основе системно-объектных моделей предметной области // *Научные ведомости Белгородского государственного университета. Серия: Экономика. Информатика*. 2013. Том 22, № 165. P. 186–194.
- Österle H. et al. Memorandum on design-oriented information systems research // *Eur. J. Inf. Syst.* 2011. Vol. 20, № 1. P. 7–10.
- Suarez-Figueroa M.C., Gomez-Perez A., Fernandez-Lopez M. The NeOn Methodology for Ontology Engineering // *Ontology Engineering in a Networked World*. 2012. P. 9–34.
- Uschold M., Gruninger M. Ontologies: Principles, Methods and Applications // *Knowl. Eng. Rev.* St Petersburg State University, 1996. Vol. 11, № 2. P. 93–136.
- Noy N.F., McGuinness D.L. Ontology Development 101: A Guide to Creating Your First Ontology // *Stanford Knowledge Systems Laboratory*. 2001. P. 25.
- Holsapple C.W., Joshi K.D. A collaborative approach to ontology design // *Commun. ACM*. 2002. Vol. 45, № 2. P. 42–47.
- Aschoff F.-R., Schmalhofer F., Van Elst L. Knowledge Mediation: A Procedure for the Cooperative Construction of Domain Ontologies // *Proceedings of the ECAI2004 Workshop on Agentmediated Knowledge Management AMKM2004*. 2004. P. 29–38.
- De Nicola A., Missikoff M., Navigli R. A Proposal for a Unified Process for Ontology Building: UPON // *International Conference on Database and Expert Systems Applications*. 2005. P. 655–664.
- De Nicola A., Missikoff M., Navigli R. A software engineering approach to ontology building // *Inf. Syst.* 2009. Vol. 34, № 2. P. 258–275.
- Vrandečić D. et al. The DILIGENT knowledge processes // *J. Knowl. Manag.* 2005. Vol. 9, № 5. P. 85–96.
- Kotis K., Vouros G. Human-centered ontology engineering: The HCOME methodology // *Knowl. Inf.* 2005. Vol. 10, № 1. P. 109–131.

22. Auer S., Herre H. RapidOWL - an Agile Knowledge Engineering Methodology // Perspectives of Systems Informatics, 6th International Andrei Ershov Memorial Conference, PSI 2006. 2006. P. 424–430.
23. Braun S., Schmidt A., Walter A. Ontology Maturing: a Collaborative Web 2.0 Approach to Ontology Engineering // World Wide Web Internet And Web Information Systems. 2007.
24. Buzan T., Buzan P. The Mind Map Book: How to Use Radiant Thinking to Maximize Your Brain's Untapped Potential. Plume, 1996.
25. Гаврилова Т.А., Гулякина Н.А. Визуальные методы работы со знаниями: попытка обзора // Искусственный интеллект и принятие решений. 2008. №1. P. 15–21.
26. Болотникова Е.С., Гаврилова Т.А., Горовой В.А. Об одном методе оценки онтологий // Известия РАН. Теория и системы управления. 2011. №3. P. 98–110.
27. Sure Y. et al. OntoEdit: Collaborative Ontology Development for the Semantic Web // Semant. Web — ISWC 2002. 2002. P. 221–235.
28. Křemen P. et al. Ontology-driven mindmapping // Proc. 8th Int. Conf. Semant. Syst. - I-SEMANTICS '12. 2012. № September 2014. P. 125.
29. Jupp S. et al. Populous: a tool for building OWL ontologies from templates // BMC Bioinformatics. 2012. Vol. 13, № Suppl 1. P. S5.
30. Rubin D.L. Creating and curating a terminology for radiology: Ontology modeling and analysis // J. Digit. Imaging. 2008. Vol. 21, № 4. P. 355–362.
31. Xiang Z. et al. Ontorat: Automatic generation of new ontology terms, annotations, and axioms based on ontology design patterns // J. Biomed. Semantics. 2015. Vol. 6, № 1. P. 1–10.
32. Fenves S.J. et al. CPM: A core model for product data // J. Comput. Inf. Sci. Eng. 2008. Vol. 8, № 1. P. 1–14.
33. <https://schema.org/Product> [Electronic resource].
34. Hepp M. GoodRelations: An ontology for describing products and services offers on the web // International Conference on Knowledge Engineering and Knowledge Management. 2008. P. 329–346.
35. Rachuri S. et al. Information sharing and exchange in the context of product lifecycle management: Role of standards // Comput. Aided Des. 2008. Vol. 40, № 7. P. 789–800.
36. UNECE. Annex II (Informative) Units of Measure: Code elements listed by name // Recommendation No. 20 - Units of Measure used in International Trade. 2005. P. 1–57.

Method for Collaborative Visual Creation of a Knowledge Graph

D. V. Kudryavtsev^I, A. M. Begler^I, T. A. Gavrilova^I, I. A. Leshcheva^I, M. V. Kubelskiy^I, O. N. Tushkanova^{II,III}

^I Graduate School of Management, St. Petersburg University, St. Petersburg, Russia

^{II} St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, St. Petersburg, Russia

^{III} InfoWings, St. Petersburg, Russia

Abstract. The paper proposes a method for creating a knowledge graph. The method is focused on the visualization techniques and group work. Necessity of the new method appeared during the project with an automotive company, which needed systematization, integration and re-use of knowledge about the assembly units of various cars within the company. Mind-maps and spreadsheets has been used during conceptualization, formalization, and implementation stages for the knowledge graph development as well as communications with experts. The need for active involvement of experts in the creation of the knowledge graph determines the scope of the method, since they and their colleagues will use the system in the future (internal customers). The created knowledge graph has a complex structure of properties associated with one class and its subclasses, while there is a little number of ontology classes. The proposed method helped to create a pilot knowledge graph, which includes more than 50 assembly units and about 400 properties.

Keywords: knowledge graph; ontology; ontology engineering; knowledge graph development, visualization techniques, collaborative development.

DOI 10.14357/20718594190103

References

1. Villazon-Terrazas B. et al. Construction of Enterprise Knowledge Graphs (I) // Exploiting Linked Data and Knowledge Graphs in Large Organisations. Springer, 2017. P. 87–116.
2. Gruber T.R. Toward principles for the design of ontologies used for knowledge sharing // Int. J. Hum. Comput. Stud. 1995. Vol. 43. P. 907–928.
3. Lenat D.B. et al. Cyc: Toward Programs With Common Sense // Commun. ACM. 1990. Vol. 33, № 8. P. 30–49.
4. Uschold M., King M. Towards a Methodology for Building Ontologies. 1995. 13 p.

5. Gruninger M., Fox M.S. The Role of Competency Questions in Enterprise Engineering // *Benchmarking — Theory and Practice*. 1995. P. 22–31.
6. Skuce D. Conventions for reaching agreement on shared ontologies // *Proceedings of the 9th Knowledge Acquisition for Knowledge Based Systems Workshop*. 1995.
7. Simperl E., Luczak-Rösch M. Collaborative ontology engineering: A survey // *Knowl. Eng. Rev.* 2014. Vol. 29, № 1. P. 101–131.
8. Fernández-López M., Gómez-Pérez A., Juristo N. METHONTOLOGY: From Ontological Art Towards Ontological Engineering // *AAAI-97 Spring Symposium Series*. 1997. Vol. SS-97-06. 33-40 p.
9. Sure Y., Staab S., Struder R. On-To-Knowledge Methodology // *Handbook on Ontologies*. 2004. P. 117–132.
10. Ohgren A., Sandkuhl K. Towards a Methodology for Ontology Development in Small and Medium-Sized Enterprises // *IADIS International Conference on Applied Computing*. 2005. P. 369–376.
11. Слободюк А.А., Маторин С.И., Четвериков С.Н. О подходе к созданию онтологий на основе системно-объектных моделей предметной области // *Научные ведомости Белгородского государственного университета. Серия: Экономика. Информатика*. 2013. Vol. 22, № 165. P. 186–194.
12. Österle H. et al. Memorandum on design-oriented information systems research // *Eur. J. Inf. Syst.* 2011. Vol. 20, № 1. P. 7–10.
13. Suarez-Figueroa M.C., Gomez-Perez A., Fernandez-Lopez M. The NeOn Methodology for Ontology Engineering // *Ontology Engineering in a Networked World*. 2012. P. 9–34.
14. Uschold M., Gruninger M. *Ontologies: Principles, Methods and Applications* // *Knowl. Eng. Rev.* St Petersburg State University, 1996. Vol. 11, № 2. P. 93–136.
15. Noy N.F., McGuinness D.L. *Ontology Development 101: A Guide to Creating Your First Ontology* // *Stanford Knowledge Systems Laboratory*. 2001. P. 25.
16. Holsapple C.W., Joshi K.D. A collaborative approach to ontology design // *Commun. ACM*. 2002. Vol. 45, № 2. P. 42–47.
17. Aschoff F.-R., Schmalhofer F., Van Elst L. Knowledge Mediation: A Procedure for the Cooperative Construction of Domain Ontologies // *Proceedings of the ECAI2004 Workshop on Agentmediated Knowledge Management AMKM2004*. 2004. P. 29–38.
18. De Nicola A., Missikoff M., Navigli R. A Proposal for a Unified Process for Ontology Building: UPON // *International Conference on Database and Expert Systems Applications*. 2005. P. 655–664.
19. De Nicola A., Missikoff M., Navigli R. A software engineering approach to ontology building // *Inf. Syst.* 2009. Vol. 34, № 2. P. 258–275.
20. Vrandečić D. et al. The DILIGENT knowledge processes // *J. Knowl. Manag.* 2005. Vol. 9, № 5. P. 85–96.
21. Kotis K., Vouros G. Human-centered ontology engineering: The HCOME methodology // *Knowl. Inf.* 2005. Vol. 10, № 1. P. 109–131.
22. Auer S., Herre H. RapidOWL - an Agile Knowledge Engineering Methodology // *Perspectives of Systems Informatics, 6th International Andrei Ershov Memorial Conference, PSI 2006*. 2006. P. 424–430.
23. Braun S., Schmidt A., Walter A. *Ontology Maturing: a Collaborative Web 2.0 Approach to Ontology Engineering* // *World Wide Web Internet And Web Information Systems*. 2007.
24. Buzan T., Buzan P. *The Mind Map Book: How to Use Radiant Thinking to Maximize Your Brain's Untapped Potential*. Plume, 1996.
25. Гаврилова Т.А., Гулякина Н.А. Визуальные методы работы со знаниями: попытка обзора // *Искусственный интеллект и принятие решений*. 2008. Vol. 1. P. 15–21.
26. Болотникова Е.С., Гаврилова Т.А., Горовой В.А. Об одном методе оценки онтологий // *Известия РАН. Теория и системы управления*. 2011. Vol. 3. P. 98–110.
27. Sure Y. et al. OntoEdit: Collaborative Ontology Development for the Semantic Web // *Semant. Web — ISWC 2002*. 2002. P. 221–235.
28. Křemen P. et al. Ontology-driven mindmapping // *Proc. 8th Int. Conf. Semant. Syst. - I-SEMANTICS '12*. 2012. № September 2014. P. 125.
29. Jupp S. et al. Populous: a tool for building OWL ontologies from templates // *BMC Bioinformatics*. 2012. Vol. 13, № Suppl 1. P. S5.
30. Rubin D.L. Creating and curating a terminology for radiology: Ontology modeling and analysis // *J. Digit. Imaging*. 2008. Vol. 21, № 4. P. 355–362.
31. Xiang Z. et al. Ontorat: Automatic generation of new ontology terms, annotations, and axioms based on ontology design patterns // *J. Biomed. Semantics*. 2015. Vol. 6, № 1. P. 1–10.
32. Fenves S.J. et al. CPM: A core model for product data // *J. Comput. Inf. Sci. Eng.* 2008. Vol. 8, № 1. P. 1–14.
33. <https://schema.org/Product> [Electronic resource].
34. Hepp M. GoodRelations: An ontology for describing products and services offers on the web // *International Conference on Knowledge Engineering and Knowledge Management*. 2008. P. 329–346.
35. Rachuri S. et al. Information sharing and exchange in the context of product lifecycle management: Role of standards // *Comput. Aided Des.* 2008. Vol. 40, № 7. P. 789–800.
36. UNECE. Annex II (Informative) Units of Measure: Code elements listed by name // *Recommendation No. 20 - Units of Measure used in International Trade*. 2005. P. 1–57.