УДК 004.89 DOI: 10.18287/2223-9537-2021-11-2-212-226

Подход к созданию онтологий на основе электронных таблиц с произвольной структурой

А.В. Видия, Н.О. Дородных, А.Ю. Юрин

Институт динамики систем и теории управления имени В.М. Матросова СО РАН, Иркутск, Россия

Аннотация

Использование семантических технологий, в том числе онтологий, является широко распространенной практикой при создании современных интеллектуальных систем. Электронные таблицы являются одним из наиболее доступных и распространенных способов представления и хранения информации, который характеризуется большим разнообразием и разнородностью компоновок, стилей и содержания, оставаясь ценным источником структурированных предметных знаний. В работе предлагается автоматизировать процесс формирования онтологий на основе анализа и преобразования электронных таблиц, обладающих произвольной структурой. Представлен подход, обеспечивающий восстановление семантики табличных данных, концептуализацию и формализацию табличного содержания в форме онтологии. Приведены основные этапы подхода и описание разработанных программных средств. Данные средства использовались при решении практической задачи формирования онтологии для диагностирования и оценки технического состояния нефтехимического оборудования. В качестве исходных данных использовались электронные таблицы, извлечённые из отчётов по экспертизе промышленной безопасности нефтехимических комплексов. По результатам апробации сделан вывод о целесообразности использования предлагаемого подхода при прототипировании предметных онтологий.

Ключевые слова: *приобретение знаний, семантические технологии, онтологический инжиниринг, онтология, электронная таблица, экспертиза промышленной безопасности*.

Цитирование: *Видия, А.В.* Подход к созданию онтологий на основе электронных таблиц с произвольной структурой / А.В. Видия, Н.О. Дородных, А.Ю. Юрин // Онтология проектирования. – 2021. – Т.11, №2(40). - С.212-226. – DOI: 10.18287/2223-9537-2021-11-2-212-226.

Введение

Семантические технологии позволяют объединить разнородную информацию из различных источников и обеспечить её представление с помощью формальных стандартизованных средств моделирования знаний. Центральным элементом семантических технологий является онтология, которая может быть использована для эффективного представления знаний и перевода знаний в форму, пригодную для интерпретации компьютерами и людьми [1]. В основном онтологии используются экспертами предметной области (ПрО), системными аналитиками и инженерами по знаниям на этапах концептуализации и формализации знаний. Построение онтологий остаётся сложной, творческой и трудоёмкой задачей. Многие годы ведутся исследования по повышению эффективности создания онтологий, начиная от разработки крупных методологий и проектов, таких как: *KACTUS* [2], *METHONTOLOGY* [3] *On-To-Knowledge* [4], *NeOn* [5], и заканчивая различными программными инструментами (например, *Protégé, Fluent Editor, OntoStudio, ONTOedit, WebOnto* и др.). Данные инструменты предоставляют широкий спектр возможности по моделированию онтологий и поддерживают основные стандарты консорциума *W3C*.

Автоматизация формирования онтологий на основе различных информационных источников, помимо экспертов (например, баз данных, текстов, концептуальных моделей), являет-

ся перспективным направлением, так как позволяет использовать большие объёмы накопленной информации, аккумулированной в различных организациях и представленной в разных форматах. В качестве такого источника могут выступать электронные таблицы (ЭТ), которые являются одним из наиболее доступных и распространённых способов представления и хранения информации. ЭТ характеризуются большим разнообразием и разнородностью компоновок, стилей и содержания, оставаясь ценным источником предметных знаний.

В работе предлагается подход для автоматизированного создания онтологий в формате *OWL2 DL* [6] на основе анализа и преобразования данных, извлечённых из ЭТ, представленных в формате *CSV*. Особенностью подхода является использование канонической (реляционной) формы для представления ЭТ, обладающих произвольной компоновкой, что обеспечивает унификацию входных данных. Подход использовался при формировании онтологии для интеллектуальной системы диагностирования и оценки технического состояния нефтехимического оборудования в рамках задачи экспертизы промышленной безопасности (ЭПБ).

1 Состояние вопроса

В настоящее время накоплен большой объём ЭТ, представленных в различных форматах: *HTML, EXCEL, CSV* и др. Современные оценки показывают, что их количество исчисляется, как минимум, сотнями миллионов [7], из которых можно извлечь миллиарды ценных фактов. Это делает ЭТ привлекательным источником знаний для различных приложений, в том числе для создания вопросно-ответных и рекомендательных систем, конструирования баз знаний, автодополнения ЭТ и др. В связи с этим два последних десятилетия активно развиваются методологические основы извлечения и преобразования данных из ЭТ.

Особый интерес представляют подходы и программные средства извлечения связанных данных и онтологий из ЭТ. В частности, существует ряд программных инструментов, позволяющих извлекать *RDF*-триплеты из ЭТ (например, *RDF123* [8], *csv2rdf4lod* [9], *Datalift* [10], *Spread2RDF* [11]), а также формировать *OWL*-онтологии (например, *Owlifier* [12], *Populous* [13], *Any2OWL* [14], *Excel2OWL* [15]). Некоторые из существующих решений включают:

- специальные предметно-ориентированные языки для описания преобразования ЭТ в наборы связанных данных или онтологии, например: *XLWrap* [16], *Mapping Master* [17], *RML* [18] и *PEARL* (плагин *Sheet2RDF* для фреймворка «*Semantic Turkey*») [19];
- языки и средства на основе расширения языка запросов *SPARQL*, в частности, *XSPARQL* [20], *SPARQL-Generate* [21] и *Tarql* [22];
- предметно-ориентированные языки, основанные на расширении различных языков программирования высокого уровня, например, *Ruby* [11].

Следует также отметить, что в данной области существует стандарт *W3C,* определяющий подход *CSVW* [23] для генерации *RDF*-документов на основе табличных данных в формате *CSV*. Также консорциум публикует список существующих конвертеров табличных данных в *RDF*-формат на сайте [24].

Несмотря на значительный прогресс в области преобразования табличных данных, существующие программные решения обладают следующими недостатками:

- Использование определённых моделей исходных ЭТ со смешанными физическими и логическими схемами. Данный факт ограничивает использование этих инструментов для обработки произвольных таблиц, представленных, например, в различных статистических отчётах. Хотя некоторые решения, основанные на применении предметно-ориентированных языков (например, *Mapping Master*, *SPARQL-Generate* и др.), предлагают достаточно гибкий механизм преобразования произвольных табличных данных, но они сложны в понимании и применении (пользователю необходимо знать синтаксис дан-

ных языков и правила построения соответствий между элементами двух форматов), а также зачастую отсутствуют средства визуальной поддержки преобразования и т.д.

- Слабая поддержка формирования полных агрегированных онтологий на основе анализа и преобразования множества ЭТ, принадлежащих к одной ПрО, т.е., как правило, средства поддерживают единичную трансформацию таблицы в онтологию (или некоторый её фрагмент) и не предполагают объединение полученных онтологий (фрагментов) в пределах обработки одного набора таблиц.
- Практически все подходы ориентированы на программистов и специалистов в области семантических технологий и не предполагают использования непрограммирующими пользователями (например, экспертами ПрО).
- Направленность средств на генерацию наборов связанных данных в формате *RDF* и слабая поддержка формата *OWL*. Как правило, в работах не уточняется, какую конкретную версию *RDF* и *OWL* поддерживает средство.
- Отсутствие проверки качества полученных онтологий и наборов связанных данных.

В статье предлагается подход к быстрому прототипированию онтологий на основе анализа и преобразования данных, извлечённых из ЭТ, обладающих произвольной структурой.

2 Предлагаемый подход

2.1 Постановка задачи

Постановку задачи можно формализовать следующим образом: необходимо определить оператор *T* преобразования произвольных ЭТ:

$$T : AS^{CSV} \rightarrow Ont^{OWL}, \tag{1}$$

где AS^{CSV} – исходная произвольная ЭТ в формате *CSV*; Ont^{OWL} – целевая онтология в формате *OWL2 DL*.

Оператор преобразования (1) может быть представлен в виде следующих этапов:

$$T = \langle T_{AS-CS}, T_{CS-OM}, T_{OM-Ont} \rangle,$$

$$T_{AS-CS} : AS^{CSV} \rightarrow CS^{CSV},\ T_{CS-OM} : CS^{CSV} \rightarrow OM,\ T_{OM-Ont} : OM \rightarrow Ont^{OWL}, \tag{2}$$

где CS^{CSV} – исходная ЭТ, представленная в реляционной форме; *OM* – онтологическая модель, представляющая собой описание ПрО на терминологическом (*TBox*) и аксиоматическом (*ABox*) уровнях; $T_{AS\text{-}CS}$ – набор правил преобразования исходной произвольной ЭТ в формате *CSV* в реляционную форму; $T_{CS\text{-}OM}$ – набор правил преобразования реляционной ЭТ в онтологическую модель; $T_{OM\text{-}Ont}$ – набор правил преобразования онтологической модели в код онтологии в формате *OWL2 DL*.

2.2 Описание этапов предлагаемого подхода

Для автоматизированного формирования онтологий на основе анализа и преобразования произвольных ЭТ предлагается подход, который может быть представлен в виде последовательности действий (см. рисунок 1).

Этап 1. Анализ и трансформация ЭТ в каноническую форму.

Задача этапа - перейти от таблиц с произвольной структурой к однотипному реляционному представлению. Данное преобразование ($T_{AS\text{-}CS}$) включает фазы: распознавание, ролевой (функциональный) и структурный анализ [25]. Для этого используется следующая структура таблицы:

$$CS^{CSV} = \{D, RH, CH\},$$

где D – блок данных, который описывает конкретные значения данных (записи), принадлежащие к одному и тому же типу данных (например, числовые, текстовые и т.д.); RH – набор заголовков строк, т.е. $RH = \{rh^1, ..., rh^n\}$, где rh^i – i-ячейка заголовка строки; CH – набор заголовков столбцов, т.е. $CH = \{ch^1, ..., ch^m\}$, где ch^j – j-ячейка заголовка столбца. Значения в ячейках блоков заголовков могут быть разделены символом «|», с помощью которого осуществляется представление иерархических отношений между заголовками (разделение заголовков на подзаголовки). Эта структура основана на представлении таблиц, предложенном в работе [26] и адаптированном для системы *TabbyXL* [27], которая используется на данном этапе.

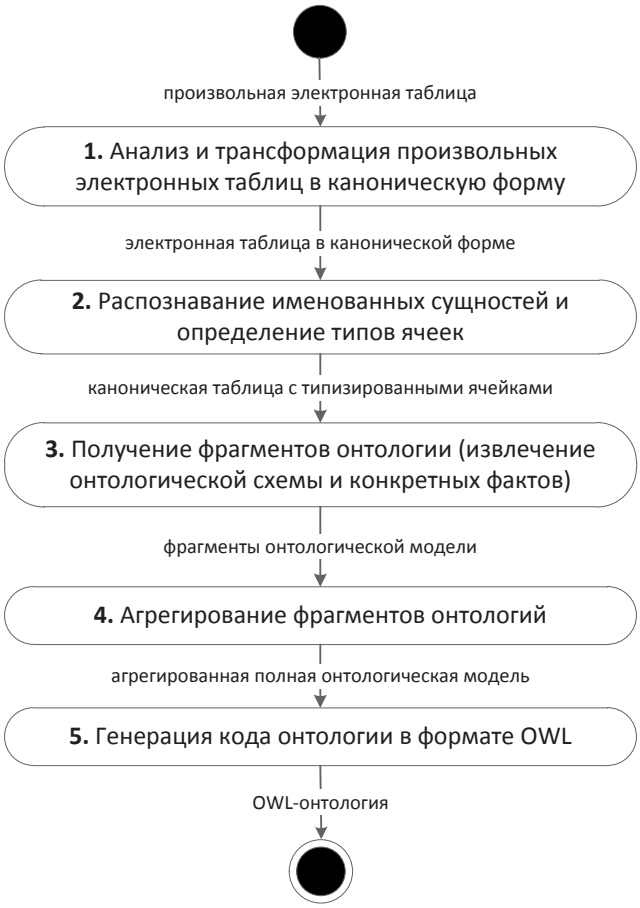


Рисунок 1 – Основные этапы предлагаемого подхода

Для реализации преобразования $T_{AS\text{-}CS}$ используется предметно-ориентированный язык *Cells Rule Language* (*CRL*) [25]. При этом набор правил может быть реализован для конкретной задачи с учётом требований к исходным и целевым данным. Был сформирован набор *CRL*-правил для двух выделенных форм таблиц, описанных в [28].

На данном этапе осуществляется подготовка данных канонической ЭТ для дальнейшей обработки, включающая:

- исправление «битых» символов Юникода;
- удаление различных «мусорных» символьных значений, кроме букв и цифр;
- расшифровка акронимов;
- удаление множественных пробелов;
- идентификация и удаление единиц измерения и т.п.

Этап 2. Распознавание именованных сущностей и определение типов ячеек.

На данном этапе осуществляется процедура извлечения и распознавания именованных сущностей, содержащихся в ячейках канонической ЭТ. Для этого используется библиотека для обработки естественного языка – *Stanford CoreNLP* и, в частности, *Java*-реализация распознавателя *Stanford Named-Entity Recognizer* (*Stanford NER*) [29]. *Stanford NER* отмечает слова в тексте, которые являются названиями объектов, такие как имена людей и компаний или наименования городов и стран. *Stanford NER* определяет множество классов именованных сущностей. В работе использовалось 8 классов (типов): *Location*, *Country*, *City*, *Person*, *Organization*, *Number*, *Percent*, *Date*. Эти классы присваиваются каждой ячейке в канонической таблице, характеризуя данные, которые она содержит. Используется неопределённый класс – *None*, который присваивается ячейке, не отнесённой ни к одному из определённых классов. Определённые типизированные ячейки подразделяются на два вида: ячейки с именованными сущностями (*named-entity cells*) и ячейки с литеральными значениями (*literal cells*). Такая типизация ячеек позволяет отнести табличные данные к разным онтологическим уровням (уровень свойств классов и конкретных экземпляров) на последующем этапе преобразования.

Этап 3. Получение фрагментов онтологии.

Задача этого этапа – получить онтологические фрагменты в виде набора классов, их отношений (объектных свойств – *object properties*) и свойств-значений (*datatype properties*), а также конкретных экземпляров (фактов – *instances*), которые описывают определённую ПрО, на основе анализа и трансформации данных канонических ЭТ. Получение фрагментов онтологии осуществляется в два последовательных шага.

На первом шаге из канонической ЭТ извлекается онтологическая схема (терминологический уровень онтологии – *TBox*) на основе заголовков (*RH* и *CH*). Анализ осуществляется построчно. В подходе используются следующие основные эвристические правила преобразования канонических ЭТ ($T_{CS\text{-}OM}$).

Правило 1: **ЕСЛИ** задан только *CH* **И** он содержит одно значение в ch^i, **ТО** ch_1^i преобразуется в класс.

Правило 2: **ЕСЛИ** задан только *CH* **И** он содержит два или более значений с разделителем («|») в ch^i, т.е. $ch^i = \{ch_1^i,...,ch_m^i\}$, **ТО** ch_1^i преобразуется в класс со свойствами-значений, соответствующими следующим значениям в множестве ch^i.

Правило 3: **ЕСЛИ** *CH* содержит одно значение в ch^i **И** оно соответствует только одному *RH*, содержащему одно значение в rh^i, **ТО** ch_1^i преобразуется в класс со свойством-значением rh_1^i.

Правило 4: **ЕСЛИ** *CH* содержит два или более значений с разделителем («|»), т.е. $ch^i = \{ch_1^i,...,ch_m^i\}$ **И** они соответствуют только одному *RH*, содержащему одно значение в rh^i, **ТО** ch_1^i преобразуется в класс со свойством-значением из rh_1^i и с дополнительными свойствами-значениями, которые соответствуют следующим значениям в множестве ch^i.

Правило 5: **ЕСЛИ** *CH* **И** *RH* содержат два или более значений с разделителем («|»), т.е. $ch^i = \{ch_1^i,...,ch_m^i\}$ **И** $rh^i = \{rh_1^i,...,rh_n^i\}$, **ТО** ch_1^i **И** rh_1^i преобразуются в соответствующие классы и указывается связь между ними.

Правило 6: **ЕСЛИ** *CH* содержит одно значение в ch^i **И** соответствует двум и более значениям *RH* с разделителем («|»), т.е. $rh^i = \{rh_1^i,...,rh_n^i\}$, **ТО** ch_1^i преобразуется в класс со свойством-значением rh_1^i, а все последующие значения из множества rh^i преобразуются в соответствующие классы и указывается связь между ними.

Подобные правила разработаны и для ситуации, где *CH* и *RH* меняются местами, т.е. структура классов формируется исходя из меток в *RH*. При этом все полученные иерархические связи заголовков интерпретируются как объектные свойства (отношения между классами). По умолчанию конкретные значения для свойств-значений устанавливаются на основе записей из *D*, которые определены на этапе 3 как литеральные.

На втором шаге из канонической ЭТ извлекаются конкретные экземпляры (аксиоматический уровень онтологии – *ABox*) на основе *D*. При этом учитываются только те значения ячеек, которые на этапе 3 были классифицированы как ячейки, содержащие именованные сущности.

Основным результатом этого этапа являются фрагменты онтологической модели. Эти фрагменты необходимо агрегировать, включая операции по уточнению названий классов, их свойств и отношений, а также их возможное слияние и разделение.

Этап 4. Агрегация фрагментов онтологии.

Задача этого этапа - объединение полученных фрагментов онтологии в единую агрегированную *OM*. Данная модель предназначена для унифицированного представления и хранения знаний, извлечённых из различных информационных источников. Модель позволяет абстрагироваться от особенностей описания знаний на различных языках и их диалектах, используемых при реализации онтологий (например, *OWL, RDFS*). Для автоматического агрегирования фрагментов *OM* используются следующие основные эвристические правила.

- Классы с одинаковыми именами объединяются, формируя общий набор свойств-значений, объектных свойств и экземпляров.
- Дублирующие классы с одинаковыми именами и структурой свойств удаляются.
- Классы с похожими именами объединяются. Полученные фрагменты *OM* могут описывать одни и те же объекты или процессы. Предлагается использовать метод сравнения строк, основанный на расстоянии Левенштейна, чтобы определить сходство между двумя именами классов. Если расстояние Левенштейна лежит в диапазоне от 0 до 3, то классы считаются подобными. Этого может быть недостаточно, поэтому учитывается структура классов (названия свойств должны частично совпадать).
- Создание новых объектных свойств (отношений между классами), если существуют одноимённые классы и свойства-значения. При этом создаётся новый класс с именем свойства-значения, а одноименное свойство-значение удаляется.
- Повторяющиеся объектные свойства между классами удаляются.
- Дублирующие свойства-значения удаляются.
- Повторяющиеся экземпляры удаляются.

Этап 5. Генерация *OWL*-кода онтологии.

Задача данного этапа - генерация кода онтологии в формате *OWL2 DL* на основе полученной *OM*. Сгенерированный *OWL*-код онтологии может быть модифицирован и дополнен с помощью различных редакторов онтологического моделирования, например, *Protégé*.

2.3 Программная реализация

Первый этап подхода реализован с использованием *TabbyXL* - консольного *Java*-приложения [27], которое обрабатывает файлы ЭТ в формате *MS Excel* (*XLSX*) или *CSV*. Каждый файл может содержать одну или несколько ЭТ с произвольной структурой. *TabbyXL* использует *CRL*-правила для преобразования данных, извлекаемых из таблиц, в каноническую форму. Преобразованные данные сохраняются в отдельных *CSV*-файлах.

Остальные этапы подхода реализованы в форме программного модуля расширения – *PKBD.Onto* [30] для системы прототипирования экспертных систем *Personal Knowledge Base Designer* (*PKBD*) [31]. *PKBD* реализован в форме настольного приложения, ориентированно-

го на непрограммирующих пользователей. Основная цель *PKBD* – это создание прототипов баз знаний, использующих формализм логических правил и онтологий. *PKBD* обладает модульной архитектурой, которая обеспечивает динамическое подключение модулей поддержки различных языков представления знаний и интеграции с инструментами концептуального и онтологического моделирования при импорте и экспорте понятий и отношений. *PKBD.Onto* представляет собой динамическую библиотеку (*Dynamic Link Library*) подключаемую через унифицированный интерфейс *API PKBD*.

3 Применение

Разработанный подход применялся в пилотном проекте Иркутского научно-исследовательского и конструкторского института химического и нефтяного машиностроения (ИркутскНИИхиммаш) при создании программного обеспечения для поддержки решения задач диагностирования и оценки технического состояния нефтехимического оборудования [32, 33]. Решаемые задачи являлись частью процедуры ЭПБ, которая заключается в подтверждении соответствия технического объекта требованиям промышленной безопасности. Анализ данной процедуры показал, что реализация этапов «разработка программы ЭПБ», «анализ и интерпретация результатов», «принятие решений по ремонту» требует обработки больших объёмов плохо формализованной информации. При этом эффективность данной обработки может быть повышена с помощью специализированных интеллектуальных систем, которые позволят:

- интерпретировать условия и параметры функционирования технических систем;
- обосновать программу технической диагностики;
- интерпретировать параметры диагностики технического состояния технических систем.

Для решения этих задач была построена онтология на основе концептуальных моделей, разработанных экспертами-предметниками, а также ЭТ, которые использовались в качестве дополнительного источника информации. Создание данной онтологии на основе анализа и преобразования ЭТ с использованием предлагаемого подхода и средств содержит следующие этапы.

Этап 1. Анализ и трансформация ЭТ в каноническую форму.

Исходные ЭТ были извлечены из шести отчётов по ЭПБ. Для этого использовались отчёты АО ИркутскНИИхиммаш, в частности, разделы, содержащие информацию о карте исходных данных по технической диагностике, результатах диагностики, расчёте прочности и остаточного ресурса. В результате анализа было извлечено 216 ЭТ, 173 из которых обладали уникальными структурой и содержанием и состояли из 5817 ячеек. Было определено два основных типа структур отобранных ЭТ.

В таблице 1 приведён пример фрагмента исходной ЭТ, извлечённой из отчёта по ЭПБ. Таблица описывает толщину стенок для основных конструктивных элементов.

В таблице 2 приведён пример фрагмента исходной ЭТ, который описывает исходные данные и результаты расчёта показателей для основных конструктивных элементов, где: Pp – давление расчётное; D – диаметр внутренний; T – температура (расчётная); [G] – допускаемое напряжение материала при расчётной температуре; Sф – толщина стенки (фактическая); Sp – толщина стенки (расчётная); [P] – допускаемое внутреннее избыточное давление.

CRL-правила были сформированы для обработки выделенных типов структур ЭТ. Таким путём были отобраны 161 ЭТ из 173 и преобразованы в каноническую форму с помощью *TabbyXL*. Оценки точности, полноты и *F*-меры составили: 0.99, 0.87 и 0.93 соответственно. *TabbyXL* показал высокую эффективность при обработке ЭТ, извлечённых из отчётов ЭПБ. «Мусорные» символы и единицы измерения были удалены из ячеек канонической ЭТ. В таб-

лицах 3 и 4 показаны фрагменты соответствующих ЭТ в каноническом виде, полученные из исходных таблиц (см. таблицы 1 и 2). Блок *D* представлен столбцом с заголовком «*DATA*», блок *CH* представлен столбцом с заголовком «*Column Heading*», а блок *RH* отсутствует.

Таблица 1 – Пример фрагмента исходной ЭТ «Толщина стенок для основных конструктивных элементов»

Конструктивный элемент	Минимальная толщина стенки по толщинометрии Smin, мм	Погрешность измерения, мм	Фактическая толщина стенки Sф, мм
Обечайка корпуса межтрубного пространства	23,7	+-0,1	23,6
Обечайка штуцера №1 ввода трубного пучка межтрубного пространства	31,7	+-0,1	31,6
Обечайка штуцера №2 ввода трубного пучка межтрубного пространства	31,7	+-0,1	31,6
Обечайка штуцера №3 ввода трубного пучка межтрубного пространства	31,7	+-0,1	31,6
Обечайка распределительной камеры №1 трубного пространства	10,0	+-0,1	9,9
Обечайка распределительной камеры №2 трубного пространства	10,0	+-0,1	9,9
Обечайка распределительной камеры №3 трубного пространства	10,0	+-0,1	9,9

Таблица 2 – Пример фрагмента исходной ЭТ «Исходные данные и результаты расчета»

Конструктивный элемент	Pp, МПа	D, мм	T, C	[G], МПа	Sф, мм	Sp, мм	[P], МПа
Обечайка корпуса межтрубного пространства	1.25	2600,0	195	142,3	23,6	12,75	2,21
Обечайка штуцера №1 ввода трубного пучка межтрубного пространства	1.25	700,0	195	148,6	31,6	3,3	11,2
Обечайка штуцера №2 ввода трубного пучка межтрубного пространства	1.25	700,0	195	148,6	31,6	3,3	11,2
Обечайка штуцера №3 ввода трубного пучка межтрубного пространства	1.25	700,0	195	148,6	31,6	3,3	11,2
Обечайка распределительной камеры №1 трубного пространства	1.2	700,0	270	140,6	9,9	3,3	3,18
Обечайка распределительной камеры №2 трубного пространства	1.2	700,0	270	140,6	9,9	3,3	3,18
Обечайка распределительной камеры №3 трубного пространства	1.2	700,0	270	140,6	9,9	3,3	3,18

Этап 2. Распознавание именованных сущностей и определение типов ячеек.

В таблицах 3b и 4b приведены фрагменты канонических ЭТ с распознанными именованными сущностями. Все ячейки данных таблиц со значением *NONE* интерпретированы как ячейки с именованными сущностями, а *NUMBER* - как ячейки с литеральными значениями.

Этап 3. Получение фрагментов онтологии.

Для формирования фрагментов онтологической схемы на основе канонических таблиц 3a и 4a были применены *Правила 1* и *2*. Следует отметить, что экземпляры были извлечены из ячеек, содержащих только именованные сущности. Список полученных экземпляров из фрагментов таблиц:

- *обечайкаКорпусаМежтрубногоПространства*;
- *обечайкаШтуцера1ВводаТрубногоПучкаМежтрубногоПространства*;
- *обечайкаШтуцера2ВводаТрубногоПучкаМежтрубногоПространства*;
- *обечайкаШтуцера3ВводаТрубногоПучкаМежтрубногоПространства*;
- *обечайкаРаспределительнойКамеры1ТрубногоПространства*;
- *обечайкаРаспределительнойКамеры2ТрубногоПространства*;
- *обечайкаРаспределительнойКамеры3ТрубногоПространства*.

Таблица 3 – Фрагменты ЭТ: a) в каноническом виде, полученной из таблицы 1; b) с распознанными сущностями

a)

DATA	Column Heading
обечайка корпуса межтрубного пространства	конструктивный элемент
23,7	минимальная толщина стенки по толщинометрии smin
+-0,1	погрешность измерения
23,6	фактическая толщина стенки sф
обечайка штуцера №1 ввода трубного пучка межтрубного пространства	конструктивный элемент
31,7	минимальная толщина стенки по толщинометрии smin
+-0,1	погрешность измерения
31,6	фактическая толщина стенки sф
обечайка штуцера №2 ввода трубного пучка межтрубного пространства	конструктивный элемент
31,7	минимальная толщина стенки по толщинометрии smin
+-0,1	погрешность измерения
31,6	фактическая толщина стенки sф
обечайка штуцера №3 ввода трубного пучка межтрубного пространства	конструктивный элемент
31,7	минимальная толщина стенки по толщинометрии smin
+-0,1	погрешность измерения
31,6	фактическая толщина стенки sф
обечайка распределительной камеры №1 трубного пространства	конструктивный элемент
10,0	минимальная толщина стенки по толщинометрии smin
+-0,1	погрешность измерения
9,9	фактическая толщина стенки sф
обечайка распределительной камеры №2 трубного пространства	конструктивный элемент
10,0	минимальная толщина стенки по толщинометрии smin
+-0,1	погрешность измерения

b)

DATA	Column Heading
NONE	NONE
NUMBER	NONE
NUMBER	NONE
NUMBER	NONE
NONE	NONE
NUMBER	NONE
NUMBER	NONE
NUMBER	NONE
NONE	NONE
NUMBER	NONE
NUMBER	NONE
NUMBER	NONE
NONE	NONE
NUMBER	NONE
NUMBER	NONE
NUMBER	NONE
NONE	NONE
NUMBER	NONE
NUMBER	NONE
NUMBER	NONE
NONE	NONE
NUMBER	NONE
NUMBER	NONE

Таблица 4 – Фрагменты ЭТ: a) в каноническом виде, полученной из таблицы 2; b) с распознанными сущностями

a)

DATA	Column Heading
обечайка корпуса межтрубного пространства	конструктивный элемент
1.25	pp
2600,0	d
195	т
142,3	g
23,6	sф
12,75	sp
2,21	p
обечайка штуцера №1 ввода трубного пучка межтрубного пространства	конструктивный элемент
1.25	pp
700,0	d
195	т
148,6	g
31,6	sф
3,3	sp
11,2	p
обечайка штуцера №2 ввода трубного пучка межтрубного пространства	конструктивный элемент
1.25	pp
700,0	d
195	т
148,6	g
31,6	sф
3,3	sp

b)

DATA	Column Heading
NONE	NONE
NUMBER	NONE
NUMBER	NONE
NUMBER	NONE
NUMBER	NONE
NUMBER	NONE
NUMBER	NONE
NUMBER	NONE
NONE	NONE
NUMBER	NONE
NUMBER	NONE
NUMBER	NONE
NUMBER	NONE
NUMBER	NONE
NUMBER	NONE
NUMBER	NONE
NONE	NONE
NUMBER	NONE
NUMBER	NONE
NUMBER	NONE
NUMBER	NONE
NUMBER	NONE
NUMBER	NONE

Остальные значения в столбце «*DATA*» были интерпретированы как конкретные значения для свойств-значений в классе «*КонструктивныйЭлемент*». Для коротких названий свойств-значений (до 3 символов) в начале названия добавляется приставка «*имеет*». Полученные фрагменты онтологии представлены на рисунке 2.

Рисунок 2 – Примеры полученных фрагментов онтологии

Этап 4. Агрегация фрагментов онтологии.

Пример агрегации полученных отдельных фрагментов онтологии в единую *ОМ* представлен на рисунке 3. Следует отметить, что при объединении классов «*КонструктивныйЭлемент*» также произошло объединение их экземпляров.

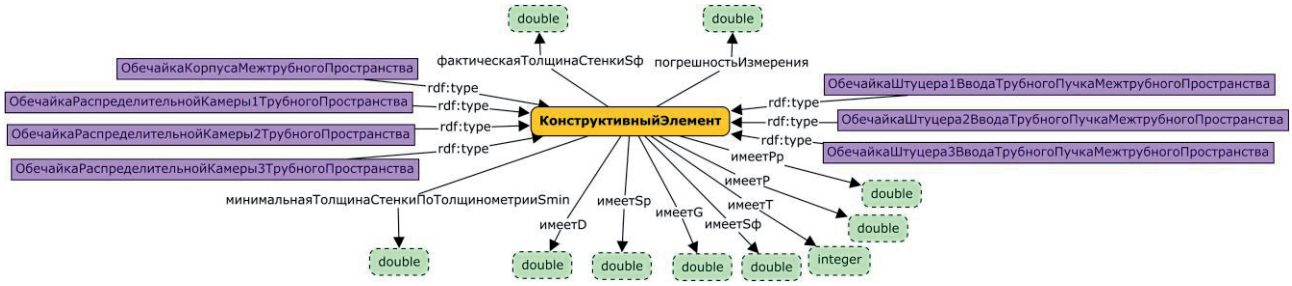


Рисунок 3 – Пример агрегированной онтологической модели на основе полученных фрагментов онтологии

Этап 5. Генерация OWL-кода онтологии.

На основе полученной *ОМ* (см. рисунок 3) была произведена генерация кода онтологии в формате *OWL2 DL*. В результате преобразования полученных канонических таблиц с помощью *PKBD* выделено: 87 классов, 32 объектных свойства (отношений между классами), 429 свойств-значений.

Заключение

В статье описан подход для автоматизированного формирования онтологий, как на уровне структуры классов (*TBox*), так и на уровне конкретных экземпляров (*ABox*). В качестве исходных данных использованы ЭТ, обладающие произвольной компоновкой и приведённые к канонической форме. Предлагаемый подход реализован в форме плагина для инструментального средства *PKBD*. Полученные *OWL*-коды являются синтаксически корректными, при этом содержательную оценку результатов должен выполнять конечный пользователь (эксперт).

Применение предлагаемого подхода осуществлено в пилотном проекте для АО ИркутскНИИхиммаш при решении задач диагностирования и оценки технического состояния нефтехимического оборудования и технологических комплексов в рамках ЭПБ. Полученные результаты показывают перспективность использования предлагаемого подхода для поддержки прототипирования онтологий.

Благодарности

Работа выполнена при финансовой поддержке Совета по грантам Президента России (проект МК-1647.2020.9).

Список источников

[1] ***Гаврилова, Т.А.*** Инженерия знаний. Модели и методы / Т.А. Гаврилова, Д.В. Кудрявцев, Д.И. Муромцев. – СПб.: Лань, 2016. – 324 с.

[2] ***Schreiber, G.*** The KACTUS View on the 'O' Word / G. Schreiber, B.J. Wielinga, W.N.H. Jansweijer // In: Proceedings of IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing, 1995. - P.159-168.

[3] ***Lopez, M.F.*** Building a chemical ontology using METHONTOLOGY and the ontology design environment / M.F. Lopez, A. Gomez-Perez, J.P. Sierra, A.P. Sierra // IEEE Intelligent Systems. - 1999. - Vol. 14(1). - P.37-46.

[4] ***Staab, S.*** Knowledge processes and ontologies / S. Staab, R. Studer, H.-P. Schnurr, Y. Sure // IEEE Intelligent Systems. - 2001. - Vol. 16(1). - P.26-34.

[5] ***Suárez-Figueroa, M.C.*** Ontology engineering in a networked world / M.C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, A. Gangemi // Springer-Verlag Berlin Heidelberg. - 2012.

[6] ***Grau, B.C.*** OWL 2: The next step for OWL / B.C. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, U. Sattler // Web Semantics: Science, Services and Agents on the World Wide Web. - 2008. - Vol. 6(4). - P.309-322.

[7] Web Data Commons. - http://webdatacommons.org.

[8] ***Han, L.*** RDF123: From spreadsheets to RDF / L. Han, T. Finin, C. Parr, J. Sachs, A. Joshi // In: Proceedings of the 7th International Semantic Web Conference (ISWC). Lecture Notes in Computer Science, vol. 5318. - Springer-Verlag Berlin Heidelberg, 2008. - P.451-466.

[9] ***Lebo, T.*** Converting governmental datasets into Linked Data / T. Lebo, G. Williams // In: Proceedings of the 6th International Conference on Semantic Systems, 2010. - P.1-3.

[10] ***Scharffe, F.*** Enabling Linked Data publication with the Datalift platform / F. Scharffe, G. Atemezing, R. Troncy, F. Gandon, S. Villata, B. Bucher, F. Hamdi, L. Bihanic, G. Képéklian, F. Cotton, J. Euzenat, Z. Fan, P.Y. Vandenbussche, B. Vatant // In: Proceedings of the AAAI workshop on semantic cities. In 26th Conference on Artificial Intelligence, W10: Semantic Cities. - 2012. - P.25-30.

[11] Spread2RDF. - https://github.com/marcelotto/spread2rdf.

[12] ***Bowers, S.*** Owlifier: Creating OWL-DL ontologies from simple spreadsheet-based knowledge descriptions / S. Bowers, J.S. Madin, M.P. Schildhauer // Ecological Informatics. - 2010. - Vol. 5(1). - P.19-25.

[13] ***Jupp, S.*** Populous: A tool for building OWL ontologies from templates / S. Jupp, M. Horridge, L. Iannone, J. Klein, S. Owen, J. Schanstra, K. Wolstencroft, R. Stevens // BMC Bioinformatics. - 2011. - Vol. 13. - P.1-12.

[14] ***Zhang, X.*** Ontology based data conversion from spreadsheet to OWL / X. Zhang, R. Di, X. Feng // In: Proceedings of the 2012 Seventh China Grid Annual Conference. - 2012. - P.76-79.

[15] ***Tahar, K.*** An Approach to support collaborative ontology construction / K. Tahar, M. Schaaf, F. Jahn, C. Kücherer, B. Paech, H. Herre, A. Winter // Studies in health technology and informatics. - 2016 - Vol. 228. - P.369-373.

[16] ***Langegger, A.*** XLWrap - Querying and Integrating Arbitrary Spreadsheets with SPARQL / A. Langegger, W. Woß // In: Proceedings of the 8th International Semantic Web Conference (ISWC). Lecture Notes in Computer Science, vol. 5823. - Springer-Verlag Berlin Heidelberg, 2009. - P.359-374.

[17] ***O'Connor, M.J.*** Mapping Master: A flexible approach for mapping spreadsheets to OWL / M.J. O'Connor, C. Halaschek-Wiener, M.A. Musen // In: Proceedings of the 9th International Semantic Web Conference (ISWC). Lecture Notes in Computer Science, vol. 6497. - Springer-Verlag Berlin Heidelberg, 2010. - P.194-208.

[18] ***Dimou, A.*** RML: A generic language for integrated RDF mappings of heterogeneous data / A. Dimou, M.V. Sande, P. Colpaert, R. Verborgh, E. Mannens, R.V.D. Walle // In: Proceedings of the Workshop on Linked Data on the Web, Co-located with the 23rd International World Wide Web Conference. - 2014. - P.1-5.

[19] ***Fiorelli, M.*** Sheet2RDF: A flexible and dynamic spreadsheet import & lifting framework for RDF / M. Fiorelli, T. Lorenzetti, M.T. Pazienza, A. Stellato, A. Turbati // In: Proceedings of the 28th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Lecture Notes in Computer Science, vol. 9101. - Springer-Verlag Berlin Heidelberg, 2015. - P.131-140.

[20] ***Bischof, S.*** Mapping between RDF and XML with XSPARQL / S. Bischof, S. Decker, T. Krennwallner, N. Lopes, A. Polleres // Journal on Data Semantics. - 2012. - Vol. 1(3). - P.147-185.

[21] ***Lefrançois, M.*** A SPARQL extension for generating RDF from heterogeneous formats / M. Lefrançois, A. Zimmermann, N. Bakerally // In: Proceedings of the 14th International Conference, European Semantic Web Conference (ESWC). Lecture Notes in Computer Science, vol. 10249. - Springer-Verlag Berlin Heidelberg, 2017. - P.35-50.

[22] Tarql: SPARQL for Tables. - http://tarql.github.io/.

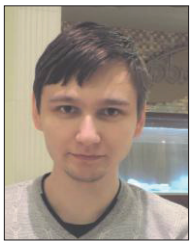
[23] Generating RDF from Tabular Data on the Web. - https://www.w3.org/TR/csv2rdf/.

[24] W3C ConverterToRdf. - https://www.w3.org/wiki/ConverterToRdf.

[25] ***Shigarov, A.O.*** Rule-based spreadsheet data transformation from arbitrary to relational tables / A.O. Shigarov, A.A. Mikhailov // Information Systems. - 2017. - Vol. 71. - P.123-136.
[26] ***Tijerino, Y.A.*** Towards ontology generation from tables / Y.A. Tijerino, D.W. Embley, D.W. Lonsdale, Y. Ding, G. Nagy // World Wide Web: Internet and Web Information Systems. - 2005. - Vol. 8(8). - P.261-285.
[27] ***Shigarov, A.O.*** TabbyXL: Software platform for rule-based spreadsheet data extraction and transformation / A.O. Shigarov, V.V. Khristyuk, A.A. Mikhailov // SoftwareX. - 2019. - Vol. 10. - 100270.
[28] ***Dorodnykh, N.O.*** Conceptual model engineering for industrial safety inspection based on spreadsheet data analysis / N.O. Dorodnykh, A.Yu. Yurin, A.O. Shigarov // In: Proceedings of the 6th International Conference on Modelling and Development of Intelligent Systems (MDIS 2019). Communications in Computer and Information Science, vol. 1126. - Springer-Verlag Berlin Heidelberg, 2020. - P.51-65.
[29] Stanford Named Entity Recognizer. - https://nlp.stanford.edu/software/CRF-NER.html.
[30] ***Дородных, Н.О.*** Разработка схем онтологий на основе преобразования электронных таблиц / Н.О. Дородных, А.В. Видия, А.Ю. Юрин // Программные продукты и системы. - 2021. - № 1. - С.124-131.
[31] ***Yurin, A.Yu.*** Personal knowledge base designer: Software for expert systems prototyping / A.Yu. Yurin, N.O. Dorodnykh // SoftwareX. - 2020. - Vol. 11. - 100411.
[32] ***Берман, А.Ф.*** Информационно-аналитическая поддержка экспертизы промышленной безопасности объектов химии, нефтехимии и нефтепереработки / А.Ф. Берман, К.А. Кузнецов, О.А. Николайчук, А.И. Павлов, А.Ю. Юрин // Химическое и нефтегазовое машиностроение. - 2018. - № 8. - С.30-36.
[33] ***Грищенко, М.А.*** Разработка интеллектуальных диагностических систем на основе онтологий / М.А. Грищенко, Н.О. Дородных, С.А. Коршунов, А.Ю. Юрин // Онтология проектирования. - 2018. - Т. 8. - № 2(28). - С.265-284. DOI: 10.18287/2223-9537-2018-8-2-265-284.

Сведения об авторах

Видия Анастасия Владимировна, 1996 г. рождения. Обучается в Институте информационных технологий и анализа данных Иркутского национального исследовательского технического университета (ИрНИТУ) по направлению Информационные системы и технологии. Программист Института динамики систем и теории управления имени В.М. Матросова СО РАН (ИДСТУ СО РАН). *Vidiya_av@mail.ru.*



Дородных Никита Олегович, 1990 г. рождения. Окончил ИрНИТУ (2012), к.т.н. (2018). Старший научный сотрудник ИДСТУ СО РАН. В списке научных трудов около 70 работ в области автоматизации создания интеллектуальных систем и баз знаний, получения знаний на основе преобразования концептуальных моделей и электронных таблиц. ORCID: 0000-0001-7794-4462; Author ID (RSCI): 979843; Author ID (Scopus): 57202323578; Researcher ID (WoS): E-8870-2014. *tualatin32@mail.ru.*

Юрин Александр Юрьевич, 1980 г. рождения. Окончил ИрНИТУ (2002), к.т.н. (2005). Заведующий лабораторией Информационных технологий исследования природной и техногенной безопасности ИДСТУ СО РАН, доцент Института информационных технологий и анализа данных ИрНИТУ. Член Российской ассоциации искусственного интеллекта и Ассоциации вычислительной техники. Член редколлегии международного научного журнала «Computer, Communication & Collaboration». В списке научных трудов более 100 работ в области разработки систем поддержки принятия решений, экспертных систем и баз знаний, использования прецедентного подхода и семантических технологий при проектировании интеллектуальных диагностических систем. ORCID: 0000-0001-9089-5730; Author ID (RSCI): 174845; Author ID (Scopus): 16311168300; Researcher ID (WoS): A-4355-2014. *iskander@icc.ru.*

Поступила в редакцию 15.05.2021, после рецензирования 16.06.21 Принята к публикации 21.06.2021.

An approach to creating freeform spreadsheet ontology

A.V. Vidia, N.O. Dorodnykh, A.Yu. Yurin

Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences (ISDCT SB RAS), Irkutsk, Russia

Abstract

The use of semantic technologies including ontologies is a widespread practice in modern intelligent system engineering. Spreadsheets are one of the most accessible and common ways of representing and storing information which are characterized by a wide variety and heterogeneity of layouts, styles and content while remaining a valuable source of domain knowledge. The paper proposes to automate the process of ontology engineering based on the analysis and transformation of spreadsheets with an arbitrary layout. For this purpose a new approach is presented that provides the restoration of the semantics of tabular data, conceptualization, and formalization of tabular content in the form of ontology. The main stages of the proposed approach and a description of the software are presented. The developed software was used to solve the practical problem of ontology engineering for diagnosing and assessing the technical condition of petrochemical equipment. Spreadsheets extracted from reports on industrial safety inspection of petrochemical complexes were used as the initial data. Based on the results of approbation, it was concluded that it is advisable to use the proposed approach when prototyping subject ontologies.

Key words: *knowledge acquisition, semantic web, ontology engineering, ontology, spreadsheet, table transformation, industrial safety inspection.*

Citation: *Vidia AV, Dorodnykh NO, Yurin AYu.* An approach to creating freeform spreadsheet ontology [In Russian]. *Ontology of designing*. 2021; 11(2): 212-226. DOI: 10.18287/2223-9537-2021-11-2-212-226.

Acknowledgment: The reported study was supported by the Council for Grants of the President of Russia (grant No. MK-1647.2020.9).

List of figures and tables

Figure 1 - The main stages of the proposed approach
Figure 2 - Examples of obtained ontology fragments
Figure 3 - An example of an aggregated ontological model based on obtained ontology fragments
Table 1 - Sample snippet of the "Wall Thickness for Major Features" source spreadsheet
Table 2 - Sample snippet of the "Initial data and calculation results" source spreadsheet
Table 3 - Fragments: a) a spreadsheet in a canonicalized form obtained from a source spreadsheet (see Table 1), b) a spreadsheet with recognized named entities
Table 4 - Fragments: a) a spreadsheet in canonicalized form obtained from a source spreadsheet (see Table 2), b) a spreadsheet with recognized named entities

References

[1] ***Gavrilova TA, Kudryavtsev DV, Muromtsev DI*** Knowledge Engineering. Models and methods [In Russian]. SPb .: Lan; 2016. 324 p.
[2] ***Schreiber G, Wielinga BJ, Jansweijer WNH*** The KACTUS View on the 'O' Word. Proc. of IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing. 1995: 159-168.
[3] ***Lopez MF, Gomez-Perez A, Sierra JP, Sierra AP*** Building a chemical ontology using METHONTOLOGY and the ontology design environment. IEEE Intelligent Systems. 1999; 14(1): 37-46.
[4] ***Staab S, Studer R, Schnurr H-P, Sure Y*** Knowledge processes and ontologies. IEEE Intelligent Systems. 2001; 16(1): 26-34.
[5] ***Suárez-Figueroa MC., Gómez-Pérez A, Motta E, Gangemi A*** Ontology engineering in a networked world. Springer-Verlag Berlin Heidelberg. 2012.
[6] ***Grau BC, Horrocks I, Motik B, Parsia B, Patel-Schneider P, Sattler U*** OWL 2: The next step for OWL. Web Semantics: Science, Services and Agents on the World Wide Web. 2008; 6(4): 309-322.

[7] Web Data Commons. Source: http://webdatacommons.org.
[8] ***Han L, Finin T, Parr C, Sachs J, Joshi A*** RDF123: From spreadsheets to RDF. Proc. of the 7th International Semantic Web Conference (ISWC). Lecture Notes in Computer Science, vol. 5318. Springer-Verlag Berlin Heidelberg, 2008: 451-466.
[9] ***Lebo T, Williams G*** Converting governmental datasets into Linked Data. Proc. of the 6th International Conference on Semantic Systems. 2010: 1-3.
[10] ***Scharffe F, Atemezing G, Troncy R, Gandon F, Villata S, Bucher B, Hamdi F, Bihanic L, Képéklian G, Cotton F, Euzenat J, Fan Z, Vandenbussche PY, Vatant B*** Enabling Linked Data publication with the Datalift platform. Proc. of the AAAI workshop on semantic cities. In 26th Conference on Artificial Intelligence, W10: Semantic Cities. 2012: 25-30.
[11] Spread2RDF. Source: https://github.com/marcelotto/spread2rdf.
[12] ***Bowers S, Madin JS, Schildhauer MP*** Owlifier: Creating OWL-DL ontologies from simple spreadsheet-based knowledge descriptions. Ecological Informatics. 2010; 5(1): 19-25.
[13] ***Jupp S, Horridge M, Iannone L, Klein J, Owen S, Schanstra J, Wolstencroft K, Stevens R*** Populous: A tool for building OWL ontologies from templates. BMC Bioinformatics. 2011; 13: 1-12.
[14] ***Zhang X, Di R, Feng X.*** Ontology based data conversion from spreadsheet to OWL. Proc. of the 2012 Seventh China Grid Annual Conference. 2012: 76-79.
[15] ***Tahar K, Schaaf M, Jahn F, Kücherer C, Paech B, Herre H, Winter A*** An Approach to support collaborative ontology construction. Studies in health technology and informatics. 2016; 228: 369-373.
[16] ***Langegger A, Woß W*** XLWrap - Querying and Integrating Arbitrary Spread-sheets with SPARQL. Proc. of the 8th International Semantic Web Conference (ISWC). Lecture Notes in Computer Science, vol. 5823. Springer-Verlag Berlin Heidelberg, 2009: 359-374.
[17] ***O'Connor MJ, Halaschek-Wiener C, Musen MA*** Mapping Master: A flexible approach for mapping spreadsheets to OWL. Proc. of the 9th International Semantic Web Conference (ISWC). Lecture Notes in Computer Science, vol. 6497. Springer-Verlag Berlin Heidelberg, 2010: 194-208.
[18] ***Dimou A, Sande MV, Colpaert P, Verborgh R, Mannens E, Walle RVD*** RML: A generic language for integrated RDF mappings of heterogeneous data. Proc. of the Workshop on Linked Data on the Web, Co-located with the 23rd International World Wide Web Conference. 2014: 1-5.
[19] ***Fiorelli M, Lorenzetti T, Pazienza MT, Stellato A, Turbati A*** Sheet2RDF: A flexible and dynamic spreadsheet import & lifting framework for RDF. Proc. of the 28th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Lecture Notes in Computer Science, vol. 9101. Springer-Verlag Berlin Heidelberg, 2015: 131-140.
[20] ***Bischof S, Decker S, Krennwallner T, Lopes N, Polleres A*** Mapping between RDF and XML with XSPARQL. Journal on Data Semantics. 2012; 1(3): 147-185.
[21] ***Lefrançois M, Zimmermann A, Bakerally N*** A SPARQL extension for generating RDF from heterogeneous formats. Proc. of the 14th International Conference, European Semantic Web Conference (ESWC). Lecture Notes in Computer Science, vol. 10249. Springer-Verlag Berlin Heidelberg, 2017: 35-50.
[22] Tarql: SPARQL for Tables. Source: http://tarql.github.io/.
[23] Generating RDF from Tabular Data on the Web. Source: https://www.w3.org/TR/csv2rdf/.
[24] W3C ConverterToRdf. Source: https://www.w3.org/wiki/ConverterToRdf.
[25] ***Shigarov AO, Mikhailov AA*** Rule-based spreadsheet data transformation from arbitrary to relational tables. Information Systems. 2017; 71: 123-136.
[26] ***Tijerino YA, Embley DW, Lonsdale DW, Ding Y, Nagy G*** Towards ontology generation from tables. World Wide Web: Internet and Web Information Systems. 2005; 8(8): 261-285.
[27] ***Shigarov AO, Khristyuk VV, Mikhailov AA*** TabbyXL: Software platform for rule-based spreadsheet data extraction and transformation. SoftwareX. 2019; 10: 100270.
[28] ***Dorodnykh NO, Yurin AYu, Shigarov AO*** Conceptual model engineering for industrial safety inspection based on spreadsheet data analysis. Proc. of the 6th International Conference on Modelling and Development of Intelligent Systems (MDIS 2019). Communications in Computer and Information Science, vol. 1126. Springer-Verlag Berlin Heidelberg, 2020: 51-65.
[29] Stanford Named Entity Recognizer. Source: https://nlp.stanford.edu/software/CRF-NER.html.
[30] ***Dorodnykh NO, Vidia AV, Yurin AYu*** Developing ontology schemas based on spreadsheet transformation. Software & Systems. 2021; 1: 124-131.
[31] ***Yurin AYu, Dorodnykh NO*** Personal knowledge base designer: Software for expert systems prototyping. SoftwareX. 2020; 11: 100411.
[32] ***Berman AF, Kuznetsov KA, Nikolaychuk OA, Pavlov AI, Yurin AYu*** Information and analytical support for the examination of industrial safety of chemical, petrochemical and oil refining facilities. Chemical and Petroleum Engineering. 2018; 8: 30-36.

[33] ***Grishchenko MA, Dorodnykg NO, Korshunov SA, Yurin AYu*** Ontology-based development of diagnostic intelligent systems [In Russian]. *Ontology of Designing*. 2018; 2(28): 265-284. DOI: 10.18287/2223-9537-2018-8-2-265-284.

About the authors

Anastasia Vladimirovna Vidia (b. 1996) a student of the Institute of information technologies and data analysis at the Irkutsk National Research Technical University (INRTU) in the direction of Information systems and technologies. She is a Programmer of Matrosov Institute for System Dynamics and Control Theory of SB RAS (ISDCT SB RAS). *Vidiya_av@mail.ru.*

Nikita Olegovych Dorodnykh (b. 1990) graduated from INRTU in 2012, PhD (2018). He is a senior associate researcher at ISDCT SB RAS. Co-author of about 70 publications in the field of computer-aided development of intelligent systems and knowledge bases, knowledge acquisition based on the transformation of conceptual models and tables. ORCID: 0000-0001-7794-4462; Author ID (RSCI): 979843; Author ID (Scopus): 57202323578; Researcher ID (WoS): E-8870-2014. *tualatin32@mail.ru.*

Alexander Yurievich Yurin (b.1980) graduated from the INRTU in 2002, PhD (2005). He is the Head of the "Information and telecommunication technologies for investigation of natural and technogenic safety" laboratory at ISDCT SB RAS and associate professor of the Institute of information technologies and data analysis of INRTU. He is a member of the Russian Association of Artificial Intelligence (RAAI) and Association for Computing Machinery (ACM). He is a member of the Editorial Board of the international scientific journal “Computer, Communication & Collaboration”. The list of scientific works includes more than 100 scientific papers in the field of development of decision support systems, expert systems and knowledge bases, application of the case-based reasoning and semantic technologies in the design of diagnostic intelligent systems, maintenance of reliability and safety of complex technical systems. ORCID: 0000-0001-9089-5730; Author ID (RSCI): 174845; Author ID (Scopus): 16311168300; Researcher ID (WoS): A-4355-2014. *iskander@icc.ru.*

Received May 15, 2021. Revised June 16, 2021. Accepted June 21, 2021.