УДК 004.89
DOI: 10.15827/0236-235X.133.124-131
Дата подачи статьи: 09.08.20
2021. Т. 34. № 1. С. 124–131

Разработка схем онтологий на основе преобразования электронных таблиц

Н.О. Дородных [1], *к.т.н., научный сотрудник, tualatin32@mail.ru*
А.Ю. Юрин [1], *к.т.н., доцент, зав. лабораторией, iskander@icc.ru*
А.В. Видия [1], *программист, vidiya_av@icc.ru*

[1] *Институт динамики систем и теории управления им. В.М. Матросова СО РАН, г. Иркутск, 664033, Россия*

Использование онтологий является широко распространенной практикой при создании интеллектуальных систем и баз знаний, в частности, для концептуализации и формализации знаний. Большинство современных подходов и инструментальных средств обеспечивают только ручное манипулирование концептами и отношениями, что не всегда эффективно. В связи с этим для автоматизированного формирования онтологий актуально использование различных информационных источников, в том числе электронных таблиц.

В данной работе описывается метод автоматизированного создания онтологических схем в формате OWL2 DL на основе анализа и преобразования данных, извлекаемых из электронных таблиц. Особенностью метода является использование для промежуточного представления электронных таблиц оригинальной канонической реляционной формы, обеспечивающей унификацию входных данных. Метод основан на принципах трансформации моделей и состоит из четырех основных этапов: преобразование исходных электронных таблиц с произвольной компоновкой в каноническую (реляционную) форму; получение фрагментов онтологической схемы; агрегация отдельных фрагментов онтологической схемы; генерация кода онтологической схемы в формате OWL2 DL.

Реализация метода осуществлена в форме двух интегрированных по данным программных средств: консольного Java-приложения TabbyXL для преобразования таблиц и модуля расширения (плагина PKBD.Onto) для системы прототипирования продукционных экспертных систем Personal Knowledge Base Designer. В качестве иллюстративного примера рассмотрено преобразование электронной таблицы с информацией о минералах, приведен результат в форме фрагмента онтологической схемы.

Метод и средства используются в учебном процессе в Институте информационных технологий и анализа данных Иркутского национального исследовательского технического университета.

Ключевые слова: *электронная таблица, каноническая таблица, концептуальная модель, онтологическая схема, OWL, трансформация моделей, генерация кода.*

Использование технологий семантического веба, в том числе онтологий [1], является широко распространенной практикой при создании интеллектуальных систем и баз знаний. В большинстве случаев онтологии используются системными аналитиками и экспертами предметной области на этапах концептуализации и формализации знаний [2]. При этом применяется различный инструментарий (например, Protégé, ONTOedit, Menthor Editor, Semaphore Ontology Editor, OntoStudio, WebOnto, Fluent Editor), который в основном обеспечивает только ручное манипулирование концептами и отношениями. Слабая интеграция подобных систем с другими информационными источниками (например, с БД, текстами, таблицами, концептуальными моделями и др.) в части импорта понятий и отношений предметной области снижает эффективность данного процесса.

В качестве источника информации для автоматизированного формирования онтологий могут выступать электронные таблицы. В настоящее время в мире циркулирует большой объем электронных таблиц, представленных в форматах HTML, XLS, XLSX, CSV [3]. Информация в данных таблицах характеризуется большим разнообразием и разнородностью компоновок, стилей, содержания, форм и форматов представления, а также высокой скоростью роста ее объема. Большой объем и свойства структуры таких таблиц делают их ценным источником в приложениях науки о данных и бизнес-аналитики. Однако, как правило, они не сопровождаются явной семантикой, необходимой для машинной интерпретации своего содержания так, как задумано их автором. Накапливаемая в таблицах информация часто является неструктурированной и нестандартизированной. Для проведения анализа

этих данных необходимы их предварительное извлечении и трансформация к структурированному представлению в соответствии с заданной формальной моделью.

В работах [4, 5] предложен подход к автоматизированному анализу и преобразованию электронных таблиц в концептуальные модели предметной области в виде диаграмм классов UML. В данном исследовании предлагается применить этот подход для создания онтологических схем (онтологий на терминологическом уровне T-Box) в формате OWL2 DL [6]. В качестве источника знаний и основы для автоматизированного формирования онтологических схем выбраны электронные таблицы, представленные в формате MS Excel, который является наиболее распространенным на сегодняшний день средством для представления структурированной информации в виде таблиц.

Таким образом, постановку задачи можно формализовать следующим образом: необходимо определить оператор T преобразования произвольных электронных таблиц:

$$T: AS^{XLSX} \rightarrow OS^{OWL}, \quad (1)$$

где AS^{XLSX} – исходная произвольная электронная таблица в формате Excel (XLSX); OS^{OWL} – целевая онтологическая схема в формате OWL2 DL.

Особенностью подхода является использование определенной канонической (реляционной) формы представления электронных таблиц, обеспечивающей унификацию входных данных.

Предлагаемый подход реализован в форме программного модуля расширения, а именно плагина PKBD.Onto для системы прототипирования продукционных экспертных систем – Personal Knowledge Base Designer (PKBD) [7]. Также рассмотрен пример применения предлагаемого подхода и модуля для создания онтологических схем в формате OWL при решении учебной задачи.

Метод разработки онтологических схем на основе электронных таблиц

Для автоматизированного формирования онтологических схем на основе анализа и трансформации произвольных электронных таблиц предлагается специализированный метод, который может быть представлен в виде последовательности действий (рис. 1).

Метод основан на принципах трансформации моделей – одного из основных понятий в области модельно-ориентированного подхода (Model-Driven Engineering) [8]. Основной особенностью разработанного метода, определяющей его новизну, является использование канонической (реляционной) формы для представления электронных таблиц, обладающих произвольной структурой.

Используя (1), подробнее рассмотрим оператор преобразования T, который с формальной точки зрения может быть представлен в виде цепочки горизонтальных экзогенных трансформаций:

$$T = (T_{AS\text{-}CS}, T_{CS\text{-}OSM}, T_{OSM\text{-}OS}),$$
$$T_{AS\text{-}CS}: AS^{XLSX} \rightarrow CS^{XLSX},$$
$$T_{CS\text{-}OSM}: CS^{XLSX} \rightarrow OSM,$$
$$T_{OSM\text{-}OS}: OSM \rightarrow OS^{OWL}, \quad (2)$$

где CS^{XLSX} – исходная электронная таблица, представленная в канонической (реляционной) форме; OSM – модель онтологической схемы; $T_{AS\text{-}CS}$ – набор правил трансформации исходной произвольной электронной таблицы в формате XLSX в каноническую форму; $T_{CS\text{-}OSM}$ – набор правил трансформации канонической электронной таблицы в модель онтологической

Рис. 1. Метод генерации онтологических схем на основе произвольных электронных таблиц

Fig. 1. A method for ontological schemas generation based on a transformation of arbitrary spreadsheets

схемы; $T_{OSM\text{-}OS}$ – набор правил трансформации модели онтологической схемы в код онтологии на терминологическом уровне T-Box в формате OWL2 DL.

Используя (2), подробнее рассмотрим основные этапы метода.

Этап 1. *Преобразование исходных электронных таблиц с произвольной компоновкой в каноническую (реляционную) форму*. Данное преобразование ($T_{AS\text{-}CS}$) включает такие фазы, как распознавание (recognition), ролевой (функциональный) и структурный анализ [9].

Используем следующую структуру канонических таблиц: $CS^{XLSX} = \{D, R^H, C^H\}$, где D – блок данных, который описывает конкретные значения данных (записи), принадлежащие к одному и тому же типу данных (например, числовые, текстовые и т.д.); R^H – набор заголовков строк; C^H – набор заголовков столбцов (значения в ячейках блоков заголовков могут быть разделены символом «|», с помощью которого осуществляется представление иерархических отношений между заголовками (разделение категорий на подкатегории)).

Эта структура основана на каноническом представлении таблиц, предложенном в [10] и адаптированном для системы TabbyXL [11], которая используется на данном этапе.

Для реализации преобразования $T_{AS\text{-}CS}$ используется предметно-ориентированный язык Cells Rule Language (CRL) [9]. При этом набор правил может быть реализован для конкретной задачи с учетом требований к исходным и целевым данным. В результате сформирован набор CRL-правил для двух выделенных форм таблиц [5]. Эти правила учитывают следующие основные случаи.

- Ячейки, расположенные в блоке заголовков столбцов и строк, могут содержать только один уровень заголовков.
- Ячейки, расположенные в блоке заголовков как столбцов, так и строк, могут содержать несколько заголовков. Эти заголовки также могут быть рассмотрены как иерархические отношения.
- Ячейки, расположенные в блоке заголовков столбцов (шапке), могут содержать несколько уровней заголовков, а ячейки, расположенные в блоке заголовков строк (заглушке), только один уровень заголовков. В этом случае заголовки столбцов могут быть интерпретированы как иерархические отношения, представляющие родительско-дочерние связи между разными категориями сущностей.
- Ячейки, расположенные в блоке заголовков строк (заглушке), могут содержать несколько уровней заголовков, а ячейки, расположенные в блоке заголовков столбцов (шапке), только один уровень заголовков. В таком случае заголовки строк могут быть интерпретированы как иерархические отношения, представляющие родительско-дочерние связи между разными категориями сущностей.

При этом осуществляется разделение каждой объединенной ячейки в исходной произвольной электронной таблице.

Этап 2. *Получение фрагментов онтологической схемы.* Основная цель этого этапа – получить шаблонные онтологические фрагменты в виде набора классов и их отношений (свойств-объектов и свойств-значений), которые описывают определенную предметную область, на основе анализа и трансформации данных из канонических электронных таблиц.

Анализ канонических электронных таблиц осуществляется построчно. При этом ячейки могут содержать несколько значений (понятий) с разделителем «|». Значение ячейки с разделителем «|» интерпретируется как иерархия либо классов (понятий), либо конкретных объектов (экземпляров классов), либо свойств. В авторском методе используются следующие основные эвристические правила преобразования ($T_{CS\text{-}OSM}$) канонических электронных таблиц.

Правило 1. ЕСЛИ R^H соответствует только одному C^H, ТО R^H преобразуется в класс со свойствами из C^H.

Правило 2. ЕСЛИ R^H соответствует только одному C^H и в то же время R^H содержит два значения с разделителем «|», то есть $R^H = (R^H - 1, R^H - 2)$, ТО $R^H - 1$ преобразуется в класс со свойствами из C^H и с дополнительным свойством *Name*, которое соответствует второму значению в $R^H - 2$.

Правило 3. ЕСЛИ R^H содержит два значения с разделителем «|» и они соответствуют двум значениям C^H с разделителем «|», ТО R^H и C^H преобразуются в соответствующие классы и указывается связь между ними.

Правило 4. ЕСЛИ R^H соответствует трем значениям C^H, то есть $C^H = (C^H - 1, C^H - 2, C^H - 3)$, с разделителем «|», ТО R^H преобразуется в класс со свойством $C^H - 1$, а $C^H - 2$ и $C^H - 3$ преобразуются в соответствующие классы и указывается связь между ними.

Подобные правила разработаны и для ситуации, когда R^H и C^H меняются местами, то есть

структура классов формируется исходя из меток в блоке заголовков столбцов C^H. При этом все полученные иерархические связи интерпретируются как объектные свойства (отношения между классами). По умолчанию значения свойств (range) устанавливаются на основе записей из блока данных *D*.

Основными результатами этого этапа являются фрагменты модели онтологической схемы *OSM*. Эти фрагменты необходимо агрегировать, включая операции по уточнению названий классов, их свойств и отношений, а также их возможное слияние и разделение.

Этап 3. *Агрегация отдельных фрагментов онтологической схемы в единую полную модель OSM*. Данная модель предназначена для унифицированного представления и хранения знаний, извлеченных из различных информационных источников. Модель позволяет абстрагироваться от особенностей описания знаний на различных языках и их диалектах, используемых при реализации онтологий (OWL, RDFS и др.).

Используя (2), подробнее опишем основные элементы *OSM*: *OSM* = (*C*, *OP*, *DP*, *DT*), где *C* – набор классов; *OP* – набор объектных свойств; *DP* – набор свойств-значений; *DT* – набор типов данных XML-схемы.

Для автоматического агрегирования фрагментов онтологической схемы используются следующие эвристические правила.

- Классы с одинаковыми именами объединяются, формируя общий набор свойств-значений и объектных свойств.
- Дублирующие классы с одинаковыми именами и структурой свойств удаляются.
- Классы с похожими именами объединяются. Полученные фрагменты модели онтологической схемы могут описывать одни и те же объекты или процессы. Предлагается использовать простой метод сравнения строк, основанный на расстоянии Левенштейна [12], чтобы определить сходство между двумя именами классов. Если расстояние Левенштейна меньше или равно 3, то будем считать классы подобными. Однако этого может быть недостаточно, так что, обращаем внимание и на структуру классов (названия свойств должны частично совпадать).
- Создание новых объектных свойств (отношений между классами), если существуют одноименные классы и свойства-значения. При этом создается новый класс с именем свойства-значения, а одноименное свойство-значение удаляется.
- Повторяющиеся объектные свойства между классами удаляются.
- Дублирующие свойства-значения удаляются.

Этап 4. *Генерация кода онтологической схемы в формате OWL2 DL на основе модели онтологической схемы* ($T_{OSM\text{-}OS}$). Данное преобразование можно описать с помощью специальных языков трансформации моделей, например, *Transformation Model Representation Language* (TMRL) [13]. В предлагаемом исследовании для реализации трансформаций авторы использовали язык программирования общего назначения. При этом все трансформации можно наглядно представить в виде следующего набора соответствий:

OSM	**OWL2 DL**
Ontology	owl:Ontology
Class	owl:Class
Class (superclass)	rdfs:subClassOf
Class (name)	rdf:about
Relationship	owl:ObjectProperty
Rhs	owl:ObjectProperty (rdfs:domain)
Lhs	owl:ObjectProperty (rdfs:range)
Property	owl:DatatypeProperty
Property (name)	owl:DatatypeProperty (rdfs:domain)
Property (value)	owl:DatatypeProperty (rdfs:range)
Property (description)	rdfs:comment

Полученный OWL-код онтологии может быть уточнен (модифицирован) с помощью различных редакторов онтологического моделирования, например Protégé и др.

Таким образом, основным результатом метода является набор классов и их свойств, которые определяют онтологическую схему на терминологическом уровне T-Box.

Программная реализация

Первый этап метода разработки онтологических схем на основе анализа и трансформации электронных таблиц реализован с использованием средства TabbyXL. Это консольное Java-приложение [11], которое обрабатывает файлы электронных таблиц в формате Excel (XLSX) или CSV. Каждый файл Excel может содержать одну или несколько электронных таблиц с произвольной структурой. TabbyXL использует CRL-правила для преобразования данных, извлекаемых из этих таблиц, в канони-

ческую форму. Преобразованные данные сохраняются в отдельных файлах.

Этапы 2–4 метода реализованы в форме программного модуля расширения (плагина) – PKBD.Onto для системы прототипирования продукционных экспертных систем PKBD [7]. PKBD реализован в форме настольного приложения, ориентированного на непрограммирующих пользователей. Основная цель PKBD – создание прототипов баз знаний, использующих формализм логических правил. PKBD обладает модульной архитектурой, которая дает возможность динамически подключать модули для поддержки различных языков представления знаний и возможность интеграции с инструментами концептуального и онтологического моделирования при импорте и экспорте понятий и отношений.

Плагин PKBD.Onto представляет собой динамическую библиотеку (Dynamic Link Library), подключаемую через унифицированный интерфейс API PKBD. Данный интерфес для поддержки модулей интеграции с внешними программами в части импорта и экспорта содержит три функции: DllInfo – получение описания подключаемой библиотеки (плагина), включая название и версию; About – получение детального описания библиотеки (плагина); Execute – выполнение основной функции библиотеки (плагина), при этом в качестве параметра передаются концептуальная модель в формате PKBD, результирующий файл, а также список возможных ключей-параметров.

В архитектуре (рис. 2) плагина PKBD.Onto можно выделить модули:

– поддержки формата концептуальных моделей PKBD, обеспечивающего доступ и манипуляцию элементами входной модели;

– преобразования входной модели в формат OWL2 DL;

– преобразования входной модели в набор связанных данных в формате RDF (может рассматриваться как средство для получения наборов конкретных фактов).

Пример применения

В настоящий момент PKBD используется в учебном процессе в Институте информационных технологий и анализа данных Иркутского национального исследовательского технического университета, поэтому в качестве примера рассмотрим решение учебной задачи разработки фрагмента онтологии. Исходными данными послужила информация о минералах (табл. 1). С целью унификации входных данных исходная электронная таблица подверглась предобработке и была представлена в виде канонической таблицы (табл. 2).

Таблица 1

Пример исходной произвольной электронной таблицы (до предварительной обработки)

Table 1

Example of an initial custom spreadsheet (before preprocessing)

Mineral	Color	Hardness (Moos)		Transparency		Syngony	
		Density	Value	Bandwidth	Refractive index	System	Appearance of crystals
Diamond	Transparent	3,47–3,55	10	Transparent	2,417–2,419	Cubic	Octahedral
Emerald	Green	2,69–2,78	7,5–8	Transparent, semi-transparent	1,56–1,6	Hexagonal	Massive
Turquoise	Blue-green	2,6–2,8	5–6	Non-transparent	1,61–1,65	Triclinic	Massive
Spinel	Red, pink	3,57–3,72	7,5–8	Transparent	1,71–1,76	Cubic	Octahedral
Chrysolite	Green	3,2–3,4	6,5–7	Transparent	1,64–1,70	Rhombic	Prismatic-pyramidal
Amethyst	Violet	2,63–2,65	7	Transparent, semi-transparent	1,543–1,554	Trigonal	Rhombohedral
Amethyst	Green	2,9–3	6–6,5	Non-transparent	1,62	Monoclinic	Massive
Topaz	Pink, yellow	3,49–3,57	8	Transparent	1,606–1,638	Rhombic	Prismatic
Opal	Yellow, red	1,96–2,2	5,5–6,5	Semi-transparent	1,44–1,46	Amorphous	Massive
Tourmaline	Red, green	3,02-3,26	7–7,5	Transparent, semi-transparent	1,616–1,652	Trigonal	Prismatic

Таблица 2

Фрагмент полученной электронной таблицы в канонической форме

Table 2

A fragment of the resulting spreadsheet in canonical form

DATA	RowHeading	ColumnHeading
Transparent	Diamond	Color
3,47-3,55	Diamond	Hardness (Moos) | Density
10	Diamond	Hardness (Moos) | Value
Transparent	Diamond	Transparency | Bandwidth
2,417-2,419	Diamond	Transparency | Refractive index
Cubic	Diamond	Syngony | System
Octahedral	Diamond	Syngony | Appearance of crystals
Green	Emerald	Color
2,69-2,78	Emerald	Hardness (Moos) | Density
7,5-8	Emerald	Hardness (Moos) | Value
Transparent, semi-transparent	Emerald	Transparency | Bandwidth
1,56-1,6	Emerald	Transparency | Refractive index
Hexagonal	Emerald	Syngony | System
Massive	Emerald	Syngony | Appearance of crystals

Далее был проведен анализ полученной канонической электронной таблицы средствами PKBD, в частности, плагином PKBD.Onto, в результате которого были извлечены фрагменты онтологической схемы. Полученные фрагменты потребовали объединения и уточнения: так, в результате применения правил агрегации все минералы объединены в класс (шаблон) Diamond, который было необходимо переименовать в Mineral.

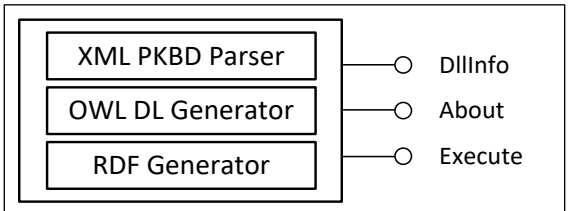

Рис. 2. Архитектура плагина PKBD.Onto

Fig. 2. A PKBD.Onto plugin architecture

После внесения в онтологическую схему необходимых изменений была выполнена генерация ее кода на ЯПЗ OWL2 DL, который может быть представлен в графическом виде (рис. 3).

Заключение

В статье описаны метод и средство в форме плагина для инструментального средства PKBD для генерации онтологических схем (онтологий на терминологическом уровне T-Box). При этом в качестве исходных данных использованы электронные таблицы в формате MS Excel (XLSX), обладающие произвольной компоновкой и приведенные к канонической форме.

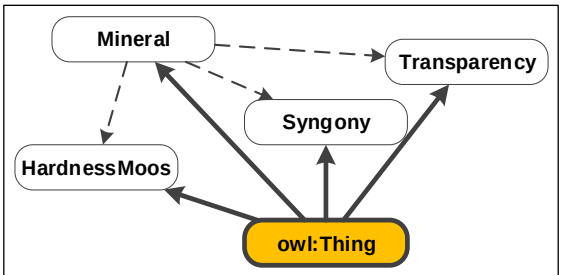

Рис. 3. Графическое представление полученной онтологической схемы (фрагмент)

Fig. 3. A graphic representation of the resulted ontological schema (a fragment)

Плагин PKBD.Onto позволяет создавать быстрые прототипы онтологий на основе электронных таблиц. Полученные OWL-коды онтологических схем являются синтаксически корректными, при этом содержательную оценку результатов должен выполнять конечный пользователь (эксперт). Доработанные и уточненные онтологии в дальнейшем можно использовать для построения интеллектуальных систем и баз знаний.

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

Литература

1. Guarino N. Formal Ontology in Information Systems. IOS Press, 1998, 348 p.
2. Гаврилова Т.А., Кудрявцев Д.В., Муромцев Д.И. Инженерия знаний. Модели и методы. СПб: Лань, 2016. 324 с.
3. Lehmberg O., Ritze D., Meusel R., Bizer C. A large public corpus of web tables con-taining time and context metadata. Proc. XXV Intern. Conf. WWW Companion, 2016, pp. 75–76. DOI: 10.1145/2872518. 2889386.
4. Dorodnykh N.O., Yurin A.Yu., Shigarov A.O. Conceptual model engineering for industrial safety inspection based on spreadsheet data analysis. Communications in computer and information science. In: Modelling and Development of Intelligent Systems, 2020, pp. 51–65.
5. Yurin A.Yu., Dorodnykh N.O. A reverse engineering process for inferring conceptual models from canonicalized tables. Proc. SIBIRCON, 2020, pp. 485–490. DOI: 10.1109/SIBIRCON48586.2019.8958458.
6. Grau B.C., Horrocks I., Motik B., Parsia B., Patel-Schneider P., Sattler U. OWL 2: The next step for OWL. SSRN Electronic Journal, 2018. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3199412. (дата обращения: 21.07.2020). DOI: 10.2139/ssrn.3199412.
7. Yurin A.Yu., Dorodnykh N.O. Personal knowledge base designer: Software for expert systems prototyping. SoftwareX, 2020, vol. 11, art. 100411. DOI: 10.1016/j.softx.2020.100411.
8. Cretu L.G., Florin D. Model-Driven Engineering of Information Systems: Principles, Techniques, and Practice. Apple Academic Press, 2014, 350 p.
9. Shigarov A.O., Mikhailov A.A. Rule-based spreadsheet data transformation from arbitrary to relational tables. Information Systems, 2017, vol. 71, pp. 123–136. DOI: 10.1016/j.is.2017.08.004.
10. Tijerino Y.A., Embley D.W., Lonsdale D.W., Ding Y., Nagy G. Towards ontology generation from tables. World Wide Web, 2005, vol. 8, no. 3, pp. 261–285. DOI: 10.1007/s11280-005-0360-8.
11. Shigarov A.O., Khristyuk V.V., Mikhailov A.M. TabbyXL: Software platform for rule-based spreadsheet data extraction and transformation. SoftwareX, 2019, vol. 10, art. 100270. DOI: 10.1016/j.softx. 2019.100270.
12. Levenshtein V.I. Binary codes capable of correcting deletions, insertions, and reversals. Dokl. Akad. Nauk SSSR, 1965, vol. 163, pp. 845–848.
13. Дородных Н.О., Юрин А.Ю. Язык для описания моделей трансформаций. Proc. ITAMS., 2018, vol. 2221, pp. 70–75. URL: http://ceur-ws.org/Vol-2221 (дата обращения: 05.08.2020).

Software & Systems
DOI: 10.15827/0236-235X.133.124-131

Received 09.08.20
2021, vol. 34, no. 1, pp. 124–131

Developing ontology schemas based on spreadsheet transformation

N.O. Dorodnykh [1], *Ph.D. (Engineering), Research Associate, tualatin32@mail.ru*
A.Yu. Yurin [1], *Ph.D. (Engineering), Associate Professor, Head of Laboratory, iskander@icc.ru*
A.V. Vidiya [1], *Programmer, vidiya_av@icc.ru*

[1] *Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences, Irkutsk, 664033, Russian Federation*

Abstract. Using ontologies is a widespread practice in the in creating intelligent systems and knowledge bases, in particular, for the conceptualization and formalization of knowledge. However, most modern approaches and tools provide only manual manipulation of concepts and relationships, which is not always effective. In this regard, using various information sources, including spreadsheets, is relevant for the automated creation of ontologies.

This paper describes a method for the automated creation of ontological schemes in the OWL2 DL format based on the analysis and transformation of data extracted from spreadsheets. A feature of the method is the use of the original canonical relational form for the intermediate representation of spreadsheets, which provides the unification of input data. The method is based on the principles of model transformation and comprises four primary stages: converting the original spreadsheets with an arbitrary layout into a canonical (relational) form; obtaining fragments of the ontological scheme; aggregation of separate fragments of the ontological scheme; generation of the code of the ontological scheme in the OWL2 DL format. The method is implemented in the form of two software tools integrated by the data: TabbyXL as the console Java application for table

conversion and the PKBD.Onto plugin as the extension module for Personal Knowledge Base Designer (software for expert systems prototyping). The transformation of a spreadsheet with information about minerals is considered as an illustrative example, and the transformation result is presented in the form of a fragment of an ontological scheme. The method and tools are used in the educational process at the Institute of Information Technologies and Data Analysis of the Irkutsk National Research Technical University (INRTU).

Keywords: spreadsheet, canonical spreadsheet, ontological schema, OWL, model transformation, code generation.

Acknowledgements. *This work was financially supported by the Council for Grants of the President of Russia, grant no. MK-1647.2020.9.*

References

1. Guarino N. *Formal Ontology in Information Systems*. IOS Press, 1998, 348 p.
2. Gavrilova T.A., Kudryavtsev D.V., Muromtsev D.I. *Knowledge Engineering. Models and Methods*. St. Petersburg, 2016, 324 p. (in Russ.).
3. Lehmberg O., Ritze D., Meusel R., Bizer C. A large public corpus of web tables con-taining time and context metadata. *Proc. XXV Intern. Conf. WWW Companion*, 2016, pp. 75–76. DOI: 10.1145/2872518.2889386.
4. Dorodnykh N.O., Yurin A.Yu., Shigarov A.O. Conceptual model engineering for industrial safety inspection based on spreadsheet data analysis. Communications in computer and information science. In: *Modelling and Development of Intelligent Systems*, 2020, pp. 51–65.
5. Yurin A.Yu., Dorodnykh N.O. A reverse engineering process for inferring conceptual models from canonicalized tables. *Proc. SIBIRCON*, 2020, pp. 485–490. DOI: 10.1109/SIBIRCON48586.2019.8958458.
6. Grau B.C., Horrocks I., Motik B., Parsia B., Patel-Schneider P., Sattler U. OWL 2: The next step for OWL. *SSRN Electronic Journal*, 2018. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3199412 (accessed July 7, 2020). DOI: 10.2139/ssrn.3199412.
7. Yurin A.Yu., Dorodnykh N.O. Personal knowledge base designer: Software for expert systems prototyping. *SoftwareX*, 2020, vol. 11, art. 100411. DOI: 10.1016/j.softx.2020.100411.
8. Cretu L.G., Florin D. *Model-Driven Engineering of Information Systems: Principles, Techniques, and Practice*. Apple Academic Press, 2014, 350 p.
9. Shigarov A.O., Mikhailov A.A. Rule-based spreadsheet data transformation from arbitrary to relational tables. *Information Systems*, 2017, vol. 71, pp. 123–136. DOI: 10.1016/j.is.2017.08.004.
10. Tijerino Y.A., Embley D.W., Lonsdale D.W., Ding Y., Nagy G. Towards ontology generation from tables. *World Wide Web*, 2005, vol. 8, no. 3, pp. 261–285. DOI: 10.1007/s11280-005-0360-8.
11. Shigarov A.O., Khristyuk V.V., Mikhailov A.M. TabbyXL: Software platform for rule-based spreadsheet data extraction and transformation. *SoftwareX*, 2019, vol. 10, art. 100270. DOI: 10.1016/j.softx.2019.100270.
12. Levenshtein V.I. Binary codes capable of correcting deletions, insertions, and reversals. *Dokl. Akad. Nauk SSSR*, 1965, vol. 163, pp. 845–848.
13. Dorodnykh N.O., Yurin A.Yu. A domain-specific language for transformation models. *Proc. ITAMS*, 2018, vol. 2221, pp. 70–75. Available at: http://ceur-ws.org/Vol-2221 (accessed August 05, 2020).

Для цитирования

Дородных Н.О., Юрин А.Ю., Видия А.В. Разработка схем онтологий на основе преобразования электронных таблиц // Программные продукты и системы. 2021. Т. 34. № 1. С. 124–131. DOI: 10.15827/0236-235X.133.124-131.

For citation

Dorodnykh N.O., Yurin A.Yu., Vidiya A.V. Developing ontology schemas based on spreadsheet transformation. *Software & Systems*, 2021, vol. 34, no. 1, pp. 124–131 (in Russ.). DOI: 10.15827/0236-235X.133.124-131.