

## Автоматизация преобразования базовых элементов ORM2-диаграммы в OWL2-онтологию

*Д.В. Литовкин, Д.С. Донцов, Е.И. Ромах, А.В. Аникин*

*Волгоградский государственный технический университет, Волгоград*

**Аннотация:** Эксперту предметной области, который не является специалистом онтологического моделирования, затруднительно построить полную и непротиворечивую OWL2-онтологию, обладающую такой степенью формализации, чтобы она могла отвечать на необходимые квалификационные вопросы (competency questions). В работе рассмотрен подход использования ORM2-диаграммы в качестве промежуточной модели для построения OWL2-онтологии. Данный подход требует использование специализированных правил преобразования элементов ORM2-диаграммы в OWL2-онтологию. Тестирование известных правил преобразования для базовых элементов ORM2-диаграммы показало, что они не соответствуют семантике нотации ORM2. Нами усовершенствованы существующие правила преобразования и автоматизирован процесс преобразования ORM2-диаграммы (состоящей из базовых элементов), в OWL2-онтологию. В результате данного исследования разработан программный компонент, позволяющий исключить ошибки преобразования и значительно уменьшить время преобразования.

**Ключевые слова:** представление знаний в явной форме, визуальная модель, промежуточная модель, онтологическое моделирование, онтология, язык OWL2, онтологический паттерн, ORM, ORM-диаграмма.

### Введение

Онтология - это фрагмент знаний о некоторой предметной области, который понятен как интеллектуальным системам, так и человеку [1]. В настоящее время наибольшей популярностью для представления онтологий пользуется язык OWL2 (Ontology Web Language) [2]. Однако, эксперту предметной области (который не является инженером онтологического моделирования) сложно создать полную и непротиворечивую OWL2-онтологию [3], обладающую такой степенью формализации, чтобы она могла отвечать на необходимые квалификационные вопросы (competency questions) [4]. Для упрощения процесса создания OWL2-онтологии применяются различные подходы, в частности, проектирование онтологии на основе концептуальной модели предметной области. Согласно этому подходу, сначала строится высокоуровневая концептуальная модель предметной

---

области, а затем элементы этой модели преобразуются в утверждения на языке OWL2 с учетом правил преобразования.

В нашей работе в качестве концептуальной модели используется ORM2-диаграмма, которая является артефактом объектно-ролевого моделирования (ORM или Object-Role Modelling) и представляет предметную область как множество объектов, играющих некоторые роли [5]. Известен подход к преобразованию элементов ORM2-диаграммы в OWL2-утверждения [6], однако применяемые правила довольно громоздки, что требует значительных трудозатрат и приводит к ошибкам преобразования.

Ранее существовал инструмент DogmaModeler [7], который позволял строить ORM2-диаграмму и преобразовывать ее в OWL2-онтологию, однако сейчас он уже не поддерживается. Существующий в настоящее время редактор ORM2-диаграмм NORMA [8] не умеет преобразовывать их в OWL2-онтологию.

Нами предлагается разработать расширение [9] для широко-используемого редактора OWL2-онтологий Protege, который бы автоматизировал построение ORM2-диаграммы и преобразование ее базовых элементов в OWL2-онтологию.

### **Материалы и методы**

В качестве промежуточной модели для построения OWL2-онтологии нами выбрана ORM2-диаграмма, которая обладает большими выразительными возможностями. Базовыми элементами ORM2-диаграммы являются следующие: класс однопольных объектов (Entity Type), характеристика объекта (Value Type), отношение «тип-подтип» между классами (Subtyping), а также унарная (Unary Role) и бинарная (Binary Role) роль, которые играют объекты классов. Пример ORM2-диаграммы, содержащей базовые элементы, приведен на рис. 1.

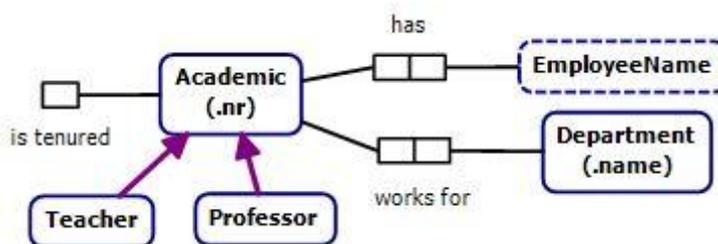


Рис. 1. – Пример ORM2-диаграммы, состоящий из базовых элементов

Для преобразования элементов ORM2-диаграммы в OWL2-онтологию мы использовали правила преобразования из [6]. Нами проведено тестирование OWL2-онтологий, порождаемых с помощью указанных правил преобразований. Тестирование показало, что семантика порождаемых OWL2-онтологий отличается от семантики исходных ORM2-диаграмм [10].

Для устранения различий в семантике OWL2-онтологий и ORM2-диаграмм нами предложены дополнительные правила преобразования для классов объектов (Entity Type), а также изменено правило для преобразования отношения “тип-подтип” (Subtyping) так, чтобы оно соответствовало строгому отношению. Новые и измененные правила преобразования представлены в таблице 1.

Нами разрабатывается Protege-расширение, которое позволяет строить ORM2-диаграмму из базовых элементов и преобразовывать ее в OWL2-онтологию с помощью усовершенствованных правил преобразования. В данной работе представлены результаты работы одного из компонентов Protege-расширения, а именно, ORM2\_OWL2\_Mapper, который выполняет преобразование ORM2-диаграммы в OWL2-онтологию. Компонент ORM2\_OWL2\_Mapper, как и само расширение, реализовано на языке Java. Как результат получен исполняемый файл с расширением jar, который помещается в папку plugins Protege-редактора.

Усовершенствованные правила преобразования базовых элементов  
ORM2-диаграммы в OWL2-онтологию

Элемент ORM2-диаграммы	Правило преобразования
Непересечение для двух несравнимых между собой классов <b>A</b> и <b>B</b> - по умолчанию, для всех классов (Entity Type) верхнего уровня	DisjointClasses(:A :B )
Реализация принципа “закрытого мира” для двух несравнимых между собой классов <b>A</b> и <b>B</b> - для всех классов (Entity Type) верхнего уровня	EquivalentClasses( owl:Thing ObjectUnionOf( :A :B ) )
классы <b>B</b> и <b>C</b> являются подтипами класса <b>A</b> - <b>строгое</b> отношение “тип-подтип” (Subtyping)	Declaration( Class( :union_of__B_C ) ) EquivalentClasses( :union_of__B_C ObjectUnionOf( :B :C ) ) Declaration( Class( :not__union_of__B_C ) ) SubClassOf( :union_of__B_C :A ) SubClassOf( :not__union_of__B_C :A ) EquivalentClasses( :A ObjectUnionOf(:union_of__B_C :not__union_of__B_C ) ) DisjointClasses( :union_of__B_C :not__union_of__B_C )

**Эксперимент, подтверждающий эффективность разработанного  
программного компонента**

Нами проведен эксперимент, целью которого является доказательство того, что с помощью разработанного компонента уменьшается время и количество ошибок преобразования ORM2-диаграммы в OWL2-онтологию по сравнению с процессом, в котором исполнителем является человек. При этом человек-исполнитель обладает знаниями о нотации ORM2-диаграммы и навыками преобразования базовых элементов диаграммы в OWL2-онтологию.

Условия эксперимента были следующие. Человеку и программному компоненту предъявлялись сходные тестовые задания. Человек получал

тестовые задания в следующей форме: 1) изображение исходной ORM2-диаграммы в файле формата .jpg; 2) OWL2-онтология, эквивалентная исходной ORM2-диаграмме (в файле формата .owl), если это предполагало тестовое задание; 3) изображение итоговой ORM2-диаграммы в файле формата .jpg; 4) перечень преобразований (в текстовой форме), которые необходимо выполнить над исходной ORM-диаграммой.

Для разработанного компонента тестовые задания формировались в виде модульных тестов.

Человек-исполнитель создавал / обновлял OWL2-онтологию, используя редактор онтологий Protege. Результаты выполнения тестового задания человеком фиксировались в следующей форме: 1) видеозапись процесса выполнения тестового задания в файле формата .mp4; 2) итоговая OWL2-онтология в файле формата .owl.

Для разработанного компонента результаты тестовых заданий проверялись посредством модульных тестов.

Всего было использовано 11 тестовых заданий. В таблице 2 показано время, затраченное человеком на выполнение тестовых заданий, а также количество совершенных им ошибок. В то время как программный компонент ORM2\_OWL2\_Mapper выполнил все тестовые задания безошибочно, затратив на выполнение каждого задания менее 1 секунды.

Результаты работы позволяют сделать следующие выводы.

1. Возможно использовать ORM2-диаграмму в качестве промежуточной модели для построения OWL2-онтологии.

2. Существуют однозначные правила преобразования базовых элементов ORM2-диаграммы в OWL2-онтологию.

3. Автоматизация процесса преобразования ORM2-диаграммы (состоящей из базовых элементов) в OWL2-онтологию посредством разработанного программного компонента ORM2\_OWL2\_Mapper позволяет

---

исключить ошибки преобразования и значительно уменьшить время преобразования.

Таблица № 2

Результаты выполнения тестовых заданий человеком

№ задания	Размер ORM2-диаграммы	Способ применения правил преобразования	Проверка OWL2-онтологии участником эксперимента	Создание / обновление OWL2-онтологии	Общее время выполнения	Общее кол-во ошибок
1	маленький	произвольный	не проводилась	создание	6 мин 10 сек	9
2	маленький	произвольный	не проводилась	обновление	5 мин 15 сек	0
3	маленький	произвольный	не проводилась	обновление	1 мин 15 сек	0
4	средний	последоват.	проводилась	создание	17 мин 15 сек	0
5	средний	последоват.	проводилась	обновление	3 мин 35 сек	0
6	средний	последоват.	проводилась	обновление	1 мин 55 сек	0
7	средний	последоват.	проводилась	обновление	10 мин 10 сек	0
8	большой	произвольный	проводилась	создание	26 мин 0 сек	16
9	большой	произвольный	проводилась	обновление	6 мин 30 сек	1
10	большой	произвольный	проводилась	обновление	5 мин 25 сек	0
11	большой	произвольный	проводилась	обновление	4 мин 45 сек	0

Литература

1. Горшков, С. Введение в онтологическое моделирование. / С. Горшков – Екатеринбург : ООО ТриниДата, 2016. – 165 с.
2. OWL 2 Web Ontology Language Document Overview (Second Edition). URL: [w3.org/TR/owl2-overview/](http://w3.org/TR/owl2-overview/) (дата обращения: 01.06.2021)
3. Мохов В.А., Сильнягин Н.Н. Интегрированный алгоритм когнитивной оценки и выбора оптимального варианта онтологической модели // Инженерный вестник Дона, 2014, №11. URL: [ivdon.ru/ru/magazine/archive/n4y2011/600](http://ivdon.ru/ru/magazine/archive/n4y2011/600).
4. Wiśniewski, D., Potoniec, J., Lawrynowicz, A., Keet, C.M.: Analysis of ontology competency questions and their formalizations in SPARQL-OWL.

Journal of Web Semantics 59, 100534 (Dec 2019).  
[doi.org/10.1016/j.websem.2019.100534](https://doi.org/10.1016/j.websem.2019.100534)

5. Halpin, T.A.: Object-Role Modeling Fundamentals: A Practical Guide to Data Modeling with ORM. Technics Publications (2015). 193 p.

6. Hodrob, R.: On Using a Graphical Notation in Ontology Engineering. Master's thesis, Birzeit University (2012). [doi.org/10.13140/RG.2.1.2812.2480](https://doi.org/10.13140/RG.2.1.2812.2480)

7. DogmaModeler: An ontology-modeling tool based on ORM. URL: [jarrar.info/Dogmamodeler/](http://jarrar.info/Dogmamodeler/) (дата обращения: 01.06.2021)

8. NORMA: A conceptual modeling tool that implements the ORM method. URL: [ormfoundation.org/files/folders/norma\\_the\\_software/default.aspx](http://ormfoundation.org/files/folders/norma_the_software/default.aspx) (дата обращения: 01.06.2021)

9. Литовкин Д.В., Аникин А.В., Сычев О.А. Онтологическое моделирование ЧТО-знаний посредством ORM2-диаграмм и разрабатываемого Protege-плагина // Инженерный вестник Дона, 2020, №12. URL: [ivdon.ru/ru/magazine/archive/n12y2020/6714](http://ivdon.ru/ru/magazine/archive/n12y2020/6714).

10. Litovkin D.V., Dontsov D.S., Anikin, A.V., Sychev, O.A. Suitability of Object-Role Modeling Diagrams as an Intermediate Model for Ontology Engineering: Testing the Rules for Mapping. BICA\*AI 2020 (Natal, Rio Grande do Norte, Brazil): Proceedings of the 11th Annual Meeting of the BICA Society, eds.: A. V. Samsonovich [et al.]. Cham (Switzerland): Springer Nature Switzerland AG, 2021. P. 188-194. URL: [link.springer.com/book/10.1007/978-3-030-65596-9](https://link.springer.com/book/10.1007/978-3-030-65596-9).

## References

1. Gorshkov S. Vvedenie v ontologicheskoe modelirovanie [Introduction to ontological modeling]. Yekaterinburg: TriniData LLC, 2016. 165 p.

2. OWL 2 Web Ontology Language Document Overview (Second Edition). URL: [w3.org/TR/owl2-overview/](http://w3.org/TR/owl2-overview/) (accessed 01/06/2021)

3. Mokhov V.A., Silnyagin N.N. Inzhenernyi vestnik Dona, 2014, №11. URL: [ivdon.ru/ru/magazine/archive/n4y2011/600](http://ivdon.ru/ru/magazine/archive/n4y2011/600).

4. Wi'sniewski, D., Potoniec, J., Lawrynowicz, A., Keet, C.M.: Analysis of ontology competency questions and their formalizations in SPARQL-OWL. *Journal of Web Semantics* 59, 100534 (Dec 2019). doi.org/10.1016/j.websem.2019.100534

5. Halpin, T.A.: *Object-Role Modeling Fundamentals: A Practical Guide to Data Modeling with ORM*. Technics Publications (2015). 193 p.

6. Hodrob, R.: *On Using a Graphical Notation in Ontology Engineering*. Master's thesis, Birzeit University (2012). doi.org/10.13140/RG.2.1.2812.2480

7. DogmaModeler: An ontology-modeling tool based on ORM. URL: jarrar.info/Dogmamodeler/ (accessed 01/06/2021)

8. NORMA: A conceptual modeling tool that implements the ORM method. URL: ormfoundation.org/files/folders/norma\_the\_software/default.aspx (accessed 01/06/2021)

9. Litovkin D.V., Anikin A.V., Sychev, O.A *Inzhenernyi vestnik Dona*, 2020, №12 URL: ivdon.ru/ru/magazine/archive/n12y2020/6714.

10. Litovkin D.V., Dontsov D.S., Anikin, A.V., Sychev, O.A. Suitability of Object-Role Modeling Diagrams as an Intermediate Model for Ontology Engineering: Testing the Rules for Mapping. *BICA\*AI 2020* (Natal, Rio Grande do Norte, Brazil): Proceedings of the 11th Annual Meeting of the BICA Society, eds.: A. V. Samsonovich [et al.]. Cham (Switzerland): Springer Nature Switzerland AG, 2021. P. 188-194. URL: link.springer.com/book/10.1007/978-3-030-65596-9.