УДК 004.822:004.75

ПОСПЕШНЫЙ А.С.

OWL 2 EL ОНТОЛОГИЯ ДЛЯ СЕМАНТИЧЕСКОГО ИНФОРМАЦИОННОГО СЕРВИСА ГРИД

Массовому внедрению онтологий и использованию интеллектуальных систем в целом препятствует тот факт, что алгоритмы логического анализа в таких системах имеют чрезвычайно высокую вычислительную сложность. Разрабатывая нашу интеллектуальную систему управления ресурсами Грид, в которой центральную роль играют онтологии, мы столкнулись с недопустимо низкой производительностью логического анализа над базами знаний с большим количеством типизированных утверждений. С целью обеспечить приемлемую скорость классификации и масштабируемость мы преобразовали нашу онтологию Грид-ресурсов к профилю OWL 2 EL и воспользовались модифицированным высокопроизводительным логическим анализатором ELK, что дало многократный прирост производительности логического анализа.

Mass deployment of ontologies and application of intelligent systems in general is hampered by the fact, that reasoning algorithms for such systems have a very high computational complexity. While developing our intellectual Grid resource management system, were ontologies play a fundamental role, we faced an unacceptably low performance of reasoning with knowledge bases that consisted of large amount of datatype expressions. In order to ensure an acceptable speed of classification and scalability, we converted our Grid resource ontology to the OWL 2 EL profile and used a modified high-performance OWL reasoner ELK, which allowed for a multiple performance gains.

Онтология Грид ресурсов

Можно с уверенностью заявить, что Грид-системы стали новым мощным и, в некоторой степени, даже необходимым инструментом для современной науки и инженерии. Однако если посмотреть на структуру любой современной Грид-системы то можно легко заметить её исключительную гетерогенность, так как все структурные составляющие Грид уникальны в отношении их программного и аппаратного обеспечения и к тому же ещё находятся в распоряжении своего непосредственного владельца.

Становится очевидным, что управление и эффективное использование Грид системы невозможно без наличия полной, достоверной и актуальной информации о всех подключенных к ней ресурсах, их характеристиках, состоянии и политике использования. Что не немаловажно, механизм доступа к этой информации обязан быть максимально понятным широкому кругу пользователей и в то же время достаточно гибким и адаптивным для широкого круга задач, так как на практике пользователями Грид могут быть специалисты из совершенно разных областей науки без глубоко понимания принципов и технологий, стоящих в основе Грид.

С целью усовершенствовать интерфейс взаимодействия пользователя с Грид-системой мы решили применить семантические технологии, развивающиеся в рамках концепции Семантической Паутины. С их помощью мы создали “интеллектуальный” информационный сервис Грид – семантический информационный сервис Grid-DL [1].

В основе такого сервиса лежит база знаний о ресурсах Грид, построенная с помощью онтологий. Пользователь может применять высокоуровневое и понятное ему описание решаемой задачи в понятных ему терминах, а необходимые для этого программные и аппаратные ресурсы будут определены и найдены при помощи системы логических выводов. Другим преимуществом такого подхода является надёжный механизм проверки корректности запроса Грид-ресурсов, так как можно проверить запрос на предмет логических несоответствий, что поможет минимизировать ошибочные назначения и простой ресурсов в связи с трудно обнаруживаемыми пользовательскими ошибками.

Упомянутая ранее онтология Грид-ресурсов является краеугольным камнем в нашей концепции семантического информационного сервиса. Будучи основанной на специальной схеме именования ресурсов Грид – GLUE (Grid Laboratory Uniform Environment) [2], эта онтология помогает фиксировать все доступные для существования в Грид компоненты и их характеристики.

Разработанная нами ранее онтология [3] содержит 65 классов, 33 объектных и 106 типизированных свойства и соответствует уровню $\mathcal{SHIF}(\mathcal{D})$ дескриптивной логики, что отвечает OWL-Lite диалекту языка OWL. Однако применив эту онтологию на практике, мы столкнулись с недопустимо низкой

производительностью логического анализа над базой знаний. Перепробовав различные алгоритмы и методы оптимизации мы пришли к выводу, что табличный алгоритм, который лежит в основе основных современных логических анализаторов (т.к. Pellet, HermiT, FaCT++), не может справиться с онтологиями большого размера, так как изначально был рассчитан на языки с большой экспрессивностью и выполняет процесс классификации онтологии путём итеративного построения модели для каждой пары концептов и поиск противоречий в ней. Согласно [4] такая задача имеет сложность $\mathcal{NE}xp\mathcal{T}ime$, что в комбинации с очень большим размером онтологии в нашем случае исключает этот подход.

За решением проблемы мы обратились к опыту разработки интеллектуальных системы на основе одних из самых больших на сегодняшний день онтологий – SNOMED CT, GALEN и Gene Ontology.

Один способ обеспечить приемлемую скорость классификации – построить онтологию на основе профиля EL языка OWL 2 [5]. Данный профиль основан на дескриптивной логике $\mathcal{EL}^{++}$ [6], что позволяет достичь необходимых вычислительных характеристик. Так для OWL 2 EL онтологий основные задачи логического анализа могут быть выполнены за полиномиальное время по отношению к размеру обрабатываемой онтологии, сохраняя при этом достаточную экспрессивность языка.

Таким образом, было решено внести необходимые изменения в разработанную ранее онтологию ресурсов Грид таким образом, что бы она соответствовала профилю EL языка OWL 2.

Адаптация онтологии к профилю EL

Согласно спецификации языка OWL 2, Таблица 1 отображает разрешённые в профиле OWL 2 EL выражения.

Таблица 1. Выражения, разрешённые профилем OWL 2 EL

Группа	Тип OWL выражений	Выражение
Определение классов	Квантор существования класса	ObjectSomeValuesFrom
	Квантор существования диапазона данных	DataSomeValuesFrom
	Квантор существования экземпляра	ObjectHasValue
	Квантор существования литерала (типизированных данных)	DataHasValue
	Самоограничение	ObjectHasSelf
	Перечисление единственного экземпляра или литерала	ObjectOneOf DataOneOf
	Пересечение именованных классов и ограничений	ObjectIntersectionOf DataIntersectionOf
Аксиомы	Утверждение подкласса	SubClassOf
	Утверждение эквивалентности классов	EquivalentClasses
	Утверждение непересекаемости классов	DisjointClasses
	Утверждения, формирующие иерархи объектных и типизированных свойств.	SubObjectPropertyOf SubDataPropertyOf
	Эквивалентность свойств	EquivalentObjectProperties EquivalentDataProperties
	Транзитивность объектных свойств	TransitiveObjectProperty
	Рефлексивность объектных свойств	ReflexiveObjectProperty
	Функциональность типизированных свойств	FunctionalDataProperty
	Область определения (домен) свойства	ObjectPropertyDomain DataPropertyDomain
	Диапазон свойства	ObjectPropertyRange DataPropertyRange
	Утверждения индивидов	SameIndividual DifferentIndividuals ClassAssertion ObjectPropertyAssertion DataPropertyAssertion NegativeObjectPropertyAssertion NegativeDataPropertyAssertion
	Ключи	HasKey

Профилем EL предусмотрено использование 19-и типов данных, большинство из которых определённо в рамках спецификации XSD – языка описания структуры XML – документов. Рисунок 1 резюмирует все поддерживаемые в рамках профиля типы данных, устанавливает их наследственность и разрешённые ограничительные аспекты (фасеты).

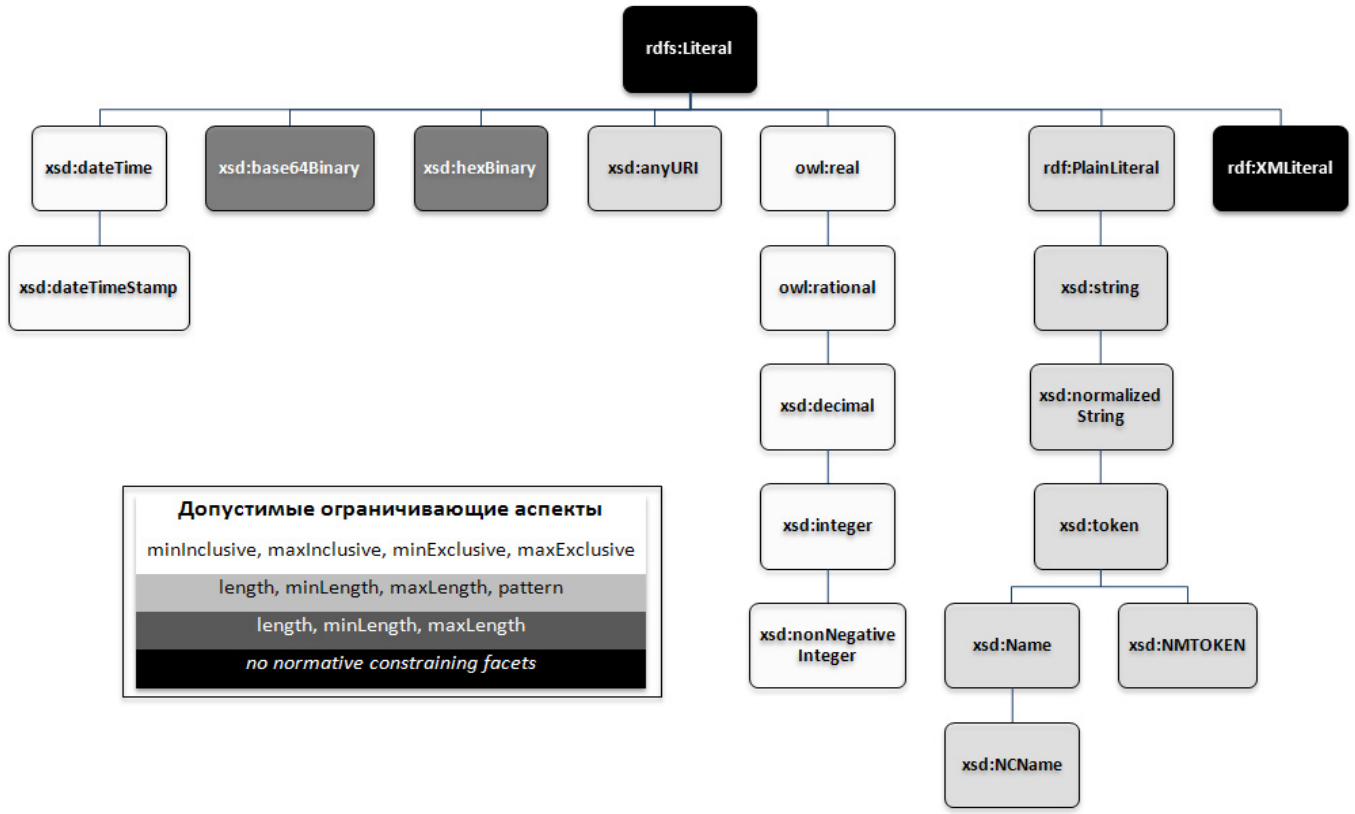

Рис 1: Допустимые типы данных в онтологиях OWL 2 EL профиля

Важно отметить, что профиль OWL 2 EL не предусматривает использование следующих типов данных: *xsd:double, xsd:float, xsd:nonPositiveInteger, xsd:positiveInteger, xsd:negativeInteger, xsd:long, xsd:int, xsd:short, xsd:byte, xsd:unsignedLong, xsd:unsignedInt, xsd:unsignedShort, xsd:unsignedByte, xsd:language и xsd:boolean.* Причина этого заключается в том, что список разрешённых профилем типов был составлен таким образом, что бы пересечение пространств значений каждого из них с любым другим из типов было либо пустым, либо бесконечным. Подобное ограничение помогает обеспечить необходимые вычислительные характеристики логического анализа EL онтологий.

Таким образом, мы смогли сопоставить выражения, используемые в нашей онтологии, с теми, которые разрешены профилем OWL 2 EL и произвели следующие модификации:

1. *Замена запрещённых типов данных.*

В нашей онтологии было использовано запрещённый в профиле EL тип данных xsd:boolean, а именно, в определении класса FileSystem (isReadOnly) и NetworkAdapter (hasInboundIP, hasOutboundIP). Данные выражения были заменены на эквивалентные им, с использование типа xsd:integer, где значение 1 соответствует булевой истине, а 0 – ложному утверждению.

2. *Замена инверсных свойств.*

В OWL онтологиях одно свойство может быть представлено как инверсия другого. Если свойство P_1 представлено как инверсия свойства P_2 и концепт X связан с Y свойством P_2, то отсюда следует, что концепт Y связан с X свойством P_1. Однако профиль OWL 2 EL запрещает использовать подобные выражения.

Данный вид свойств широко используется в нашей онтологии ресурсов Грид, так как помогает использовать более понятные и удобно читаемые выражения. Так, следующие свойства объявлены как инверсные друг другу:

describes	⇆	describedBy
hasServiceData	⇆	isServiceDataFor
hasSoftwareData	⇆	isSoftwareDataFor
hasVOInfo	⇆	isVOInfoFor
hasVOView	⇆	isVOViewFor
managedBy	⇆	manages
partOf	⇆	contains

Для решения этой проблемы мы изъяли из онтологии утверждения инверсности указанных выше свойств и внесли изменения в алгоритм наполнения базы знаний таким образом, что бы при формировании онтологии создавались новые утверждения между экземплярами, связанные упомянутыми объектными свойствами. То есть, если экземпляр *A* и *B* соединены свойством describes, то будет добавлено новое утверждение о том, что *B* describedBy *A*.

3. *Замена функциональных свойств.*

Если свойство объявлено как функциональное, то оно имеет не более одного значения для каждого экземпляра. Эта характеристика известна как наличие уникального свойства.

В нашей онтологии 11 из 32 объектных свойств объявлены как функциональные. К сожалению, профиль OWL 2 EL не предусматривает их использование, поэтому мы вынуждены были извлечь их из онтологии (был удалён признак функциональности, сами объектные свойства остались частью онтологии).

Что бы всё-таки обеспечить проверку уникальности, были внесены изменения в алгоритм наполнения онтологии таким образом, что бы проверка уникальности и консистентности происходила на этом этапе. Таким образом, обеспечивается то же поведение, что и наличие функциональных свойств в онтологии.

4. *Замена обратно-функциональных свойств.*

Если свойство обратно-функционально, то инверсия этого свойства функциональна. То есть, инверсия данного свойства имеет не больше одного значения для каждого индивида. Эта характеристика также известна как недвусмысленное (однозначное) свойство.

Подобные конструкции запрещены к использованию в OWL 2 EL. В нашей онтологии всего одно свойство объявлено как обратно-функциональное – hasLocation. Таким образом, можно было отразить тот факт, что если разные Грид-площадки (Site) расположены в одном и том же месте, то либо это одна и та же площадка, либо это ошибочная запись.

Как и в предыдущем случае, данную проверку, симулирующую поведение обратно-функциональных свойств, было перенесено на этап формирования базы знаний.

5. *Замена симметричных свойств.*

Последнее несоответствие нашей онтологии с профилем EL было наличие симметрических свойств. Если объектное свойство *P* объявлено симметричным, то при объявлении *P(x,y)* автоматически подразумевается *P(y,x)*. В нашей онтологии симметричным являться свойство inRelationshipWith, с помощью которого устанавливаются связи между сервисами Грид (если сервис *A* связан с сервисом *B*, то следовательно сервис *B* тоже связан с *A*).

Аналогично инверсным свойствам, при формировании онтологии создаются новые утверждения между экземплярами, связанные упомянутым симметричным свойством.

Таким образом, после перечисленных выше модификаций, наша онтология ресурсов Грид стала соответствовать профилю OWL 2 EL, не потеряв при этом своих эксплуатационных качеств. Теперь у нас появилась возможность воспользоваться высокопроизводительными логическими анализаторами для EL онтологий в нашем семантическом информационном сервисе Грид, многократно повысив его производительность.

Адаптация онтологии к логическому анализатору ELK

Модифицировав онтологию нашего семантического информационного сервиса Грид таким образом, что бы она соответствовала профилю EL языка OWL 2, мы приступили к выбору наиболее подходящего логического анализатора.

Были рассмотрены такие логические анализаторы как CB, CEL, jCEL, Snoroket и ELK. Последний, благодаря своей продуманной масштабируемой архитектуре стал в основу нашей работы.

ELK является свободно распространяемым высокопроизводительным логическим анализатором с открытым исходным кодом для онтологий OWL 2 EL профиля. Исключительная высокая производительность логического анализа достигается за счёт применения оптимизированного параллельного накопительного алгоритма для рассуждений, работа которого, в общих чертах, заключается в итеративном применении к множеству исходных аксиом трансформирующих правил, порождающих новые аксиомы- следствия.

EKL обладает гибкой модульной архитектурой и может быть использован в разнообразных конфигурациях, в частности возможно использование ELK как утилиты, вызываемой из командной строки операционной системы, как библиотеки, интегрируемой в приложение посредством интерфейса OWL API или как подключаемого расширения к редактору онтологий и баз знаний Protege.

Исчерпывающее описание алгоритмов используемых в ELK, особенности их реализации и доказательства их корректности и полноты можно найти в работах [7-8].

Однако, несмотря на то что ELK позиционирует себя как логический анализатор для OWL 2 EL онтологий, ним поддерживаются далеко не все перечисленные ранее выражения этого профиля.

Самим критическим недостатком ELK было отсутствие поддержки типизированных свойств и выражений с их участием. Так как наша онтология в основном хранит знания о характеристиках Грид-ресурсов именно с помощью типизированных свойств, подобное упущение было недопустимо.

Потому нами была разработана модификация логического анализатора ELK которая добавила полноценную поддержку логического анализа онтологий с типизированными свойствами. Более детально об этих модификациях можно узнать в работе [9].

Другим, менее критическим недостатком было отсутствие поддержки области определения и диапазона свойств (Domain и Range). Природа этого ограничения незначительно влияет на эксплуатационные характеристик нашей системы, однако мы всё же внесли изменения в алгоритм формирования базы знаний таким образом, что бы производить предварительную проверку экземпляров на принадлежность к указанным классам. В скором времени, новая версия ELK будет уже иметь полноценную поддержку упомянутых выражений и данный шаг уже будет не нужен.

Тестирование

Для проверки работоспособности наших модификаций мы воспользовались генератором баз знаний Грид ресурсов и создали несколько онтологий разного размера и сложности.

Наши тестовые базы знаний содержат информацию о различных элементах Грид-системы, которая обслуживает выполнение экспериментов на Большом адронном коллайдере. Онтология представляет собой терминологический блок и утверждения конкретных экземпляров ресурсов Грид и отношений между ними.

В качестве «полезной нагрузки» мы добавили к онтологии несколько определений, который будут выполнять роль пользовательского поискового запроса к базе знаний, а именно, мы будем искать подходящую Грид-площадку расположенную в Соединённом Королевстве, которая на данный момент простаивает и содержит кластер x86_64 архитектуры с узлами, в которых бьём оперативной памяти находится в пределах от 4 до 8 Гб:

```
UK_Site ≡ Site and hasLocation
some (Location and hasName some
  string[pattern ".*, UK"])

Idle_CE ≡ ComputingElement and
  hasState some (CEState and
  hasRunningJobs value 0 and
  hasWaitingJobs value 0 and
hasFreeJobSlots some integer[>0])
 and hasState some (CEState and
  hasStatus value Production)

x64HightMemCluster ≡ SubCluster
    and (describedBy some
    (hasPlatformType value
"x86_64"^^string) and (describedBy
 some (hasRAMSize some integer
    [>= 4096, <= 8192])))

MySite ≡ UK_Site and (contains
some IdleCE) and (contains some
    x64HightMemCluster)
```

По результатам классификации, экземпляры класса MySite будут представлять искомые ресурсы.

В таблице 2 приведены результаты тестирования. Тестовая конфигурация: Intel Core 2 Duo T9300 @ 2.50GHz, 4 Gb RAM, OpenSUSE 12.1 (Linux 3.1.10), Java 1.7.0_05 (-Xmx3200M -Xms3200M). Каждый тест проводился 3 раза. Использовалась обычная модифицированная сборка ELK, без дополнительных оптимизаций (в некоторых случаях возможно достичь двукратного прироста производительности, без потери корректности полученных результатов).

Таблица 2. Результаты тестирования

Конфигурация онтологии	Параметры онтологии	Время классификации, мс
Грид-площадки	13018 аксиом, 2300 экземпляров, 1916 объектных утверждений, 5979 типизированных утверждений.	2 790
Грид-площадки + Грид-сервисы	234729 аксиом, 36367 экземпляров, 68362 объектных утверждений, 93090 типизированных утверждений.	8 062
Грид-площадки + Грид-сервисы + Вычислительные элементы	519653 аксиом, 73900 экземпляров, 110401 объектных утверждений, 260861 типизированных утверждений.	11 776
Грид-площадки + Грид-сервисы + Вычислительные элементы + Кластеры + Подкластеры	822520 аксиом, 80824 экземпляров, 128633 объектных утверждений, 531630 типизированных утверждений.	43 729
Грид-площадки + Грид-сервисы + Вычислительные элементы + Кластеры + Подкластеры + Хранилища данных + Политики безопасности	1117487 аксиом, 133029 экземпляров, 192408 объектных утверждений, 658397 типизированных утверждений.	55 621

Заключение

Полученные результаты позволяют нам с оптимизмом смотреть на перспективы использования семантического информационного сервиса Грид на практике. Модификация базовой онтологий и применение перспективного высокопроизводительного логического анализатора ELK, адаптированного нами для поддержки типизированных выражений, позволило многократно повысить производительность нашей системы и свести время обработки пользовательского запроса к приемлемым величинам.

Описанная онтология доступна для свободной загрузки по адресу http://grid-ontology.googlecode.com/files/GLUE_EL.owl

Программа, производящая формирование базы знаний на основе информационной системы Грид доступна по адресу https://grid-ontology.googlecode.com/svn/trunk/Importer

Список литературы

1. Поспешный А.С., Стиренко С.Г. GRID-DL – семантический информационный сервис ГРИД // Компьютинг, 2011. – Том 10, Выпуск 3. – С. 285 – 294.
2. S.Andreozzi, S. Burke, F. Ehm, L. Field, et al. GLUE Schema Specification v. 2.0. – 2009.
3. Поспешный А.С., Стиренко С.Г. Онтология ресурсов для семантического информационного сервиса Грид, Электронное моделирование. 33 (4) (2011).
4. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. The Description Logic Handbook: Theory, Implementation, and Applications. 2nd ed. Cambridge University Press, 2007
5. OWL 2 Web Ontology Language Profiles. W3C Recommendation 27 October 2009. http://www.w3.org/TR/owl2-profiles
6. F. Baader, S. Brandt, and C. Lutz. Pushing the EL Envelope Further. In Proceedings of the OWLED 2008 Workshop on OWL: Experiences and Directions, 2008.
7. Yevgeny Kazakov, Markus Krötzsch, František Simančík. Concurrent Classification of EL Ontologies**.** Proceedings of the 10th International Semantic Web Conference (ISWC-11). LNCS 7032, Springer, 2011.
8. Yevgeny Kazakov, Markus Krötzsch, František Simančík. The Incredible ELK: From Polynomial Procedures to Efficient Reasoning with EL Ontologies. http://elk.semanticweb.org/papers/elk-journal-2013.pdf
9. Поспешний А.С., Стиренко С.Г. Эффективный алгоритм рассуждений для OWL 2 EL онтологий с типизированными выражениями на базе логического процессора ELK. // Адаптивные системы автоматического управления. – 2012. – № 20 (40). – С. 106-115.