

*Хала К.О.**Міжнародний науково-навчальний
центр інформаційних технологій і систем
НАН та МОН України, м. Київ
ceceroongreat@ukr.net*

За останні роки, багато формальних онтологій були запропоновані в якості вирішення проблем опису для складних сфер знань. Добре продумані онтології мають ряд позитивних моментів в тому числі, 1) можливість визначати керовані словники термінів, 2) здатність успадковувати і розширювати існуючі властивості, 3) можливість заявити про взаємозв'язки між існуючими властивостями і 4) можливість включити нові відносини, на основі міркувань з існуючих термінів. За допомогою технологій, відомих під загальною назвою Semantic Web, особливо мови OWL [1], дослідники можуть поширювати і ділитися онтологіями по всьому науковому співтовариству. Хоча й існує ряд високоякісних онтологій, вчені далекі від реалізації всіх переваг їх використання, ще існує можливість для значного прогресу в цій області, особливо в застосуванні формальних міркувань.

Унікальна сила формальних онтологій в області подання знань є їх здатність бути акцентованими на логічні докази. Такі міркування здійснюються з використанням дескрипційної логіки (DL), форми логіки, розробленої для міркувань про об'єкти, як окремо так і про класи об'єктів. Програмне забезпечення під назвою резонер (HermiT, Pellet або Fact ++)) використовує правила DL для виконання конкретних операцій над базами знань [1]. Найбільш важливими з них є: 1) перевірка узгодженості: об'єднання онтологічної моделі з правилами DL; 2) перевірка здійсненості: здатність для описаних класів, реалізуватися реальними екземплярами; і 3) класифікація: розширення відносин між об'єктами, що були виведені з відносин в явному вигляді. Мова OWL пропонує багатий набір властивостей, але незважаючи безліч реляційних властивостей різновидів мов OWL, вони не охоплює весь спектр виражальних можливостей для відносин об'єктів, які можна побудувати.

Для таких випадків і є корисною спеціалізована мова правил Semantic Web Rule Language (SWRL) є поєднанням мови правил (RuleML) з OWL [2]. Правила визначаються як: апіорні і апостеріорні. Якщо всі оператори в попередньому висловлюванні визначаються як правдиві, то всі твердження в подальшому висловлюванні застосовні. Таким чином, нові властивості можуть бути присвоєні екземплярам, в онтології, заснованій на поточному стані бази знань. SWRL також визначає бібліотеку вбудованих функцій, які можуть застосовуватися до екземплярів. Вони включають в себе чисельні порівняння, прості арифметичні дії і маніпуляцію з рядками, часові функції. На даний момент SWRL є найбільш широко використовуваною мовою правил в співтоваристві Semantic Web.

Таке популярне середовище розробки онтологій як Protégé включає в себе плагін SWRLTab, для створення і обробки правил SWRL [2,3]. SWRL підтримується резонером Pellet до місця, де правила можуть бути визначені як «DL-safe».

Protégé [4] являє собою гнучку платформу, що налаштовується для розробки довільних моделей керованих додатків і компонентів. Вона має відкриту архітектуру, яка дозволяє програмістам інтегрувати плагіни, які можуть з'являтися у вигляді окремих вкладок, специфічних компонентів інтерфейсу користувача (віджети), або виконувати будь-які інші завдання на поточній моделі.

Protégé-OWL API [5] є відкритим вихідним кодом бібліотеки Java для OWL і RDF(S). API надає класи і методи для завантаження і збереження файлів OWL, запитів і маніпулювання моделями OWL даних, а також для виконання міркувань на основі механізму DL. Крім того, API оптимізовано для реалізації користувальницьких графічних інтерфейсів.

Jena [6] є одним з найбільш широко використовуваних Java API, для RDF і OWL, надаючи послуги для представлення моделі, синтаксичного аналізу, наполегливості бази даних, виконання запитів і деяких інструментів візуалізації. Protégé-OWL API (v 3.4) і більш низькі версії інтегровані з Jena, а аналізатор Jena ARQ використовується парсером Protégé-OWL.

Інтерфейси моделі Protégé-OWL розташовані в ієрархії успадкування. Огляд доступних інтерфейсів можна знайти у [1], з базовим інтерфейсом всіх RDF ресурсів, з яких отримані підінтерфейси для класів, властивостей і примірників (об'єктів).

Існує чітке розходження в моделі між пойменованими класами і анонімними класами. Пойменовані класи використовуються для створення окремих екземплярів, в той час як анонімні класи використовуються для визначення логічних характеристик (обмежень) з названих класів. Класи, що логічно визначені, можуть бути використані для побудови складних виразів з обмежень класу і пойменованих класів. Як і обмеження, логічні класи мають сенс тільки, якщо вони приєднані до певного пойменованих класу або властивості.

Багато процесів в області знань часто краще моделювати з використанням декларативного підходу і правил, що призводить до зацікавленості в системах, заснованих на правилах. Проте, можливість взаємодії серед безлічі існуючих систем, заснованих на правилах обмежена. Технологія SWRL [2] стала першим кроком, так як заснована на поєднанні OWL з правилами Markup Language [7]. Поєднання OWL і SWRL надає можливість проведення логічного доказу за межами можливостей класифікації вбудованих в опис логік, реалізованих OWL.

Розширення SWRL долають складні сценарії, які включають в себе математичні співвідношення і формули, які перевищують поточні можливості SWRL, що були запропоновані [8, с.358].

SWRL семантика заснована на OWL DL, так що не підтримує прямих міркувань про класи і властивості. Правило SWRL містить попередню частину, яка згадується як тіло, і наступну частину, яка згадується в якості голови. І тіло і голова складаються з позитивних кон'юнкція атомів:

$$atom \wedge atom \dots \implies atom \wedge atom$$

У той час як SWRL не підтримує заперечення атомів або заперечення як відмова або диз'юнкція, він підтримує класичне заперечення. Наприклад, програміст (?P) являє собою атом, де програміст є ім'ям OWL класу, а ?P є змінна, що представляє собою OWL індивідуум. Неформально правило SWRL може бути прочитано, що, якщо всі атоми в передуючій частині правдиві, то, наступна частина, також повинна бути правдивою. Існує сім типів атомів, постійного виду $P(\text{arg1}, \text{arg2}, \dots, \text{argh})$, тобто предикат P і його аргументи:

- атоми класу
- атоми властивостей індивідів
- атоми властивостей, що передаються за значеннями, даних
- атоми різних індивідів
- атоми подібних індивідів
- вбудовані атоми
- атоми діапазону даних

У SWRL, предикатні символи можуть включати в себе класи OWL, властивості або типи даних. Аргументи можуть бути окремі екземпляри або значення даних OWL, або змінні, пов'язані з ним. Всі змінні в SWRL розглядаються як квантори загальності, з їх обмеженням обсягу даного правила.

Вбудовані SWRL є предикатами, які визнають, що беруться дані одного або декількох оцінюваних аргументів. Ряд основних вбудованих функцій для математичних і строкових операцій містяться в SWRL Proposal [9]. Ці вбудовані модулі визначені в файлі `swrlb.owl` [10]. За узгодою, всім базовим SWRL може передувати специфікатор імені простору `swrlb`.

У статті були наведені переваги використання SWRL, що на даний момент є найбільш широко використовуваною мовою правил в співтоваристві Semantic Web. Описано можливості Protégé-OWL API, оскільки популярне середовище розробки онтологій Protégé включає в себе плагін SWRLTab, для створення і обробки правил SWRL. Було проаналізовано, що SWRL підтримується резонерами Protégé до місця, де правила можуть бути визначені як «DL-safe».

Література

1. World Wide Web technical report and recommendation for OWL <https://www.w3.org/TR/owl-guide/>
2. World Wide Web technical report and recommendation for SWRL <https://www.w3.org/TR/swrl-guide/>
3. SWRLTab Protégé plug-in <http://protege.cim3.net/cgi-bin/wiki.pl?SWRLTab>
4. Protégé <http://protege.stanford.edu/>

5. Protégé-OWL AP <http://protege.stanford.edu/plugins/owl/api/>
6. Jena <http://jena.sourceforge.net/>
7. Rule Markup Language <http://www.ruleml.org>
8. Sánchez-Macián, A., Pastor, E., Vergara, J. de L., & López, D. (2007). Extending SWRL to Enhance Mathematical Support. *Web Reasoning and Rule Systems* (p. 358–360). Retrieved from http://dx.doi.org/10.1007/978-3-540-72982-2_30
9. <http://www.daml.org/rules/proposal/builtins.html>
10. <http://www.w3.org/2003/11/swrlb>

Анотація. У статті наведені переваги використання SWRL, що на даний момент є найбільш широко використовуваною мовою правил в співтоваристві Semantic Web. Описано можливості Protégé-OWL API, оскільки популярне середовище розробки онтологій Protégé включає в себе плагін SWRLTab, для створення і обробки правил SWRL. А ще середовище Protégé надає SWRL підтримку вбудованих резонерів.

Ключові слова: OWL; Markup Language; SWRL; доказова аргументація; резонер; система придбання знань; редактор онтології Protege

Аннотация. В статье приведены преимущества использования SWRL, что на данный момент является наиболее широко используемым языком правил в сообществе Semantic Web. Описаны возможности Protégé-OWL API, поскольку популярная среда разработки онтологий Protégé включает в себя плагин SWRLTab, для создания и обработки правил SWRL. А еще среда Protégé предоставляет SWRL поддержку встроеным резонеров.

Ключевые слова: OWL; Markup Language; SWRL; доказательная аргументация; резонер; система приобретения знаний; редактор онтологии Protege.

Resume. The article describes the advantages of using SWRL, which is currently the most widely used language in the community rules Semantic Web. Possibilities Protégé-OWL API, because the popular development environment for ontologies Protégé includes a plugin SWRLTab, for creating and processing SWRL rules. And Protégé environment provides support SWRL with build-in reasoner.

Keyword: OWL; Markup Language; SWRL; case based reasoning; reasoned; knowledge acquisition system; Protege ontology editor