

Nowe metody reprezentacji reguł dla Sieci Semantycznej

Weronika T. Furmańska¹, Grzegorz J. Nalepa¹

¹ Katedra Automatyki, Akademia Górniczo-Hutnicza.
E-mail: wtf@agh.edu.pl, gjn@agh.edu.pl

Streszczenie. Projekt Sieci Semantycznej ma na celu dostarczenie metod efektywnego opisywania i interpretowania wiedzy w sieci Internet w oparciu o metody strukturalizacji danych i tworzenia adnotacji semantycznych. W artykule omówiono podstawowe zagadnienia reprezentacji wiedzy w projekcie Sieci Semantycznej, ze szczególnym uwzględnieniem reprezentacji reguł. Zagadnienie reprezentacji reguł jest teraz obszarem intensywnych prac i najistotniejszych problemów badawczych. W artykule rozważana jest integracja w ramach Sieci Semantycznej metody reprezentacji reguł zaczerpniętej z regułowych systemów ekspertowych XTT rozwijanej w projekcie HeKatE.

Słowa kluczowe. Sieć Semantyczna, reprezentacja reguł, XTT

1 Wprowadzenie

Główną ideą Sieci Semantycznej (ang. *Semantic Web*) jest reprezentowanie w globalnej sieci Internetu danych i powiązań między nimi w sposób umożliwiający maszynom nie tylko ich składowanie i prezentację, ale i „rozumienie” ich znaczenia, automatyczne przetwarzanie, integrację i wymianę danych pomiędzy aplikacjami [1,2].

Celem *Semantic Web* jest stworzenie środowiska, w którym programy agentowe, poruszając się po stronach internetowych, wykorzystywaliby powiązania między danymi pochodzącymi z różnych źródeł, a identyfikowanymi przez unikalne adresy URI, do pozyskiwania wiedzy i rozwiązywania skomplikowanych zadań wyszukiwania, planowania itp. Aby tego dokonać konieczne jest stworzenie uniwersalnego formatu reprezentowania danych zrozumiałego dla maszyn, a także sposobu ich interpretowania oraz języka reguł umożliwiającego wnioskowanie na ich temat.

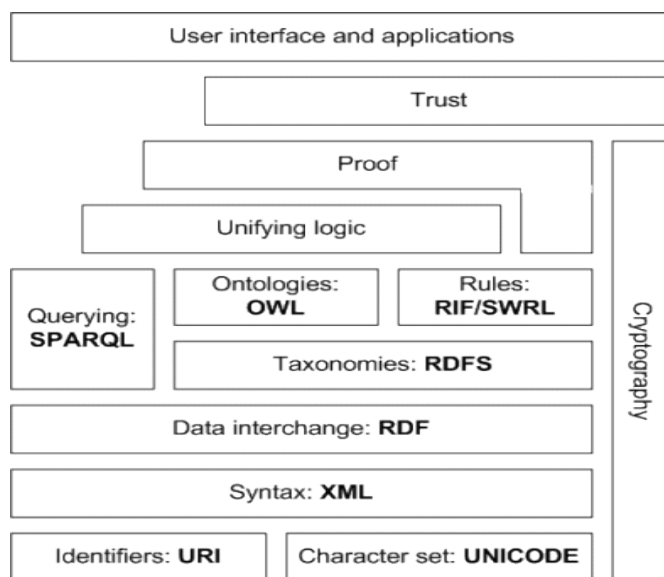
Semantic Web czerpie z dorobku metod reprezentowania i przetwarzania wiedzy, lecz ze względu na specyfikę Internetu, stawia przed tą dziedziną nowe wyzwania. Języki *Semantic Web* nie mogą być dostosowane do wąskiej grupy danych i reguł stosowanych w pojedynczym systemie. Powinny zapewniać możliwość konwersji różnych specyficznych danych i reguł do Sieci Semantycznej. Stosowana logika musi być z jednej strony wystarczająco złożona, aby reprezentować najróżniejsze powiązania między danymi, z drugiej zaś możliwa do stosowania przez programy agentowe.

Specyfika Internetu zmusza badaczy do rezygnacji z koncepcji ustrukturalizowanej i jednolitej architektury na rzecz rozwiązania rozproszonego [3].

W chwili obecnej podstawowym wyzwaniem stojącym przed badaczami jest opracowanie metod i formatów ekspresywnego a równocześnie efektywnego obliczeniowo reprezentowania i przetwarzania reguł decyzyjnych. Rozpatrywany jest zarówno opis logiczny z wykorzystaniem Logik Opisowych (ang. *Description Logics*) jak i jego serializacji i wymiana w formacie opartym o język XML.

W artykule omówione są wybrane metody reprezentacji wiedzy regułowej i kierunki badań w tym obszarze. Przedstawiona jest również nowa metoda reprezentowania reguł dla regułowych systemów ekspertowych. Zaproponowana jest koncepcja autorskiego rozwiązania, które pozwoliłoby na efektywną integrację tego podejścia z metodami Sieci Semantycznej.

Układ artykułu jest następujący: w części 2. omówione są podstawowe elementy architektury Sieci Semantycznej, a następnie w części 3. zaprezentowane wybrane języki tworzenia ontologii. Dalej, w części 4. omówione są najistotniejsze podejścia związane z opisem reguł. Część 5. przedstawia najważniejsze elementy reprezentacji wiedzy regułowej w projekcie HeKatE. W części 6. zaproponowane jest podejście do integracji tej metody z projektem Sieci Semantycznej. Propozycje metod ewaluacji przedstawione są w części 7. Na zakończenie omówione są kierunki przyszłych prac.



Rysunek 1. Stos semantyczny [2].

2 Architektura Sieci Semantycznej

Technologie Sieci Semantycznej często przedstawiane są w postaci tzw. stosu semantycznego, czyli *semantic stack* albo *layer cake* (patrz Rys. 1) [2].

Wszystkie dane reprezentowane w Sieci Semantycznej określane są mianem zasobów i identyfikowane za pomocą unikalnych *URI - Universal Resource Identifiers*. Każdy identyfikator URI składa się ze znaków standardu Unicode. W kolejnej warstwie znajduje się format *XML - eXtensible Markup Language* umożliwiający zapis danych w sposób ustrukturalizowany. XML pozwala użytkownikowi definiować własne znaczniki określające strukturę i elementy danych, jednakże znaczenie tych struktur nie jest znane zewnętrznym aplikacjom. Semantyka zapisywana jest za pomocą podstawowego języka Semantic Web tj. *RDF - Resource Description Framework*. Język ten koduje zależności pomiędzy danymi za pomocą tzw. "trójek" (ang. *triples*), odpowiadających prostym zdaniom w języku naturalnym, zawierającym podmiot, orzeczenie i dopełnienie. Trójki te zapisywane są za pomocą XML-a. Dokument RDF zawiera stwierdzenia w postaci: dany *zasób* posiada *własność* o następującej *wartości*. Każdy element trójki jest identyfikowany przez URI, zarówno podmiot, dopełnienie, jak i orzeczenie. Identyfikacja podmiotu i dopełnienia pozwala wielu dokumentom odwoływać się jednoznacznie do tych samych zasobów. Z kolei identyfikacja własności zapewnia ścisłą definicję każdej z nich. Ponadto umożliwia tworzenie nowych zależności pod warunkiem umieszczenia ich definicji w sieci pod unikalnym adresem. Trójki RDF tworzą grafy - sieć informacji o powiązanych zasobach [3,4].

Z uwagi na rozproszoną naturę Internetu i wielość baz danych, na których ma operować Semantic Web, pojawia się problem z integracją informacji pochodzących z wielu źródeł. Ta sama zależność może być opisana w różny sposób w różnych systemach. Dlatego kolejnym podstawowym komponentem Sieci Semantycznej są *ontologie*. Ontologia w Semantic Web, oznacza plik formalnie definiujący relacje pomiędzy wyrażeniami. Typowa ontologia składa się z tzw. taksonomii, czyli sposobu klasyfikacji termów, ich systematyki, oraz zbioru reguł wnioskowania. Taksonomia definiuje klasy obiektów oraz relacje pomiędzy nimi. Możliwe jest zdefiniowanie własności danej klasy, a także dziedziczenia klas. Znaczenie wyrażenia dokumentu Semantic Web jest definiowane przez ontologię, do której odnosi się dany dokument. Może się jednak zdarzyć, że dwa dokumenty opisujące to samo wyrażenie, odwołują się do dwóch różnych ontologii definiujących to wyrażenie inaczej. Konieczne zatem jest wprowadzenie w ontologiach tzw. relacji równoważności, mówiącej, że dane definicje są równoważne.

Kolejną warstwą Semantic Web jest poziom logiki, a w nim uniwersalny język służący do wyrażania informacji i reguł wnioskowania, zapisanych w ontologiach. Programy agentowe, wykorzystując ten język mogłyby przeprowadzać wnioskowanie według ściśle określonych reguł, czyli inaczej przeprowadzać dowody zadanych celów.

Wizja Sieci Semantycznej to środowisko Internetu, w którym działają inteligentne programy agentowe, przeprowadzające wnioskowanie na podstawie informacji zawartych w różnych źródłach i wymieniające się wiedzą. Takie agenty mogłyby działać jak konkretne usługi należące do różnych instytucji, np. banków, firm, centrów medycznych. Aby wnioskowanie było wiarygodne oraz by uniknąć nadużyć, każdy agent powinien mieć swój podpis elektroniczny.

3 Mechanizmy reprezentowania ontologii

W ogólnym przypadku ontologia formalna pozwala na klasyfikację pojęć. Co za tym idzie, reprezentuje informacje o pojęciach i relacjach pomiędzy nimi, oraz pozwala na realizację podstawowych zadań wnioskowania co do tych pojęć. W praktycznym modelowaniu ontologii wykorzystywane są oparte o XML języki modelowania pojęć i relacji, w tym prostszy RDF/S oraz silniejszy OWL. W przypadku OWL-a formalny opis ontologii realizowany w oparciu o logiki opisowe (*Description Logics*).

RDF to język służący do opisu danych zawartych na stronie internetowej w sposób sformalizowany. Umożliwia on współpracę różnych aplikacji Sieci Semantycznej. RDF zapisuje informacje o danych w formie trójek: zasób, własność, wartość. Poszczególne elementy identyfikowane są przez adresy URI. Aby uniknąć każdorazowego wpisywania długich URI stosuje się przestrzenie nazw. RDFS czyli RDF Schema to rozszerzenie RDF umożliwiające definiowanie klas, własności i relacji między nimi. RDFS pozwala na tworzenie w RDF klas i własności specyficznych dla danej aplikacji. Mechanizmy te pozwalają na tworzenie prostych ontologii, których przeszukiwanie możliwe jest za pomocą języka zapytań SPARQL.

OWL (*Ontology Web Language*) to język umożliwiający tworzenie ontologii i manipulowanie informacjami. Umożliwia rozumowanie oparte na *Open-World Assumption*, w przeciwieństwie do języków Prolog czy SQL. Oznacza to, że jeżeli twierdzenie nie może być udowodnione, nie można z tego wyciągnąć wniosku, że jest nieprawdziwe. Poprzednikiem OWL był język DAML+OIL, który z kolei powstał w wyniku połączenia języków DAML (DARPA Agent Markup Language) i OIL (Ontology Inference Layer lub Ontology Interchange Language). OWL dzieli się na trzy podjęzyki, różniące się stopniem złożoności i siłą wyrazu.

- OWL Lite - najprostsza wersja języka, umożliwia zapis hierarchii pojęć i podstawowych ograniczeń. Stosunkowo prosta do migracji słowników i taksonomii, mniej złożona pod względem formalnym.
- OWL DL (Description Logic) - zawiera konstrukcje języka, które odpowiadają wyrażeniom Description Logics. Mogą one być używane tylko przy spełnieniu pewnych restrykcji (np. klasa może dziedziczyć po kilku klasach, ale nie może być instancją kilku klas). Ograniczenia zapewniają, że każdy możliwy do wyprowadzenia wniosek z danej bazy wiedzy będzie wyprowadzony w skończonym czasie.
- OWL Full - jest najbardziej uniwersalnym formatem i gwarantuje największą swobodę w tworzeniu ontologii. Przykładowo klasa może być jednocześnie traktowana jako jednostka oraz zbiór jednostek. OWL Full umożliwia ontologiom rozszerzanie predefiniowanych wyrażeń RDF i OWL. Ceną ekspresyjności OWL Full jest brak gwarancji otrzymania wyniku (wniosku).

Za pomocą OWL możemy definiować klasy i relacje między nimi, a także opisywać cechy poszczególnych własności (properties) takie jak: zwrotność, przechodniość itp. OWL jest nadbudowany nad RDF i RDFS, serializowany do XML-a. Wykorzystuje znaczniki z przestrzeni nazw RDF, RDFS i OWL.

Logi Opisowe (ang. *Description Logics*, DL) są rodziną formalizmów stanowiących między innymi podstawę języka OWL. Wywodzą się one od sieci semantycznych (ang. *semantic networks*) oraz ram (faset, ang. *frames*), w przeciwieństwie do nich zapewniają jednak metody sformalizowanego opisu i wnioskowania. Czerpiąc z rachunku predykatów, mają znacznie większą siłę ekspresji niż rachunek zdań. Z drugiej strony silne ograniczenia, polegające m.in. na użyciu predykatów wyłącznie z dwoma zmiennymi logicznymi, pozwalają na konstruowanie efektywnych obliczeniowo narzędzi wnioskujących. Podstawowe terminy DL to pojęcia, instancje (indywidua) i role. Baza wiedzy składa się z dwóch podstawowych części: TBox i ABox, które opisują odpowiednio terminologię (taksonomię pojęć) i stwierdzenia (asercje) o instancjach. Wersja DL nazywana *SHOIN* jest wykorzystywana do opisu OWL.

Na bazie ontologii budowane są reguły. Do ich opisu służą opisane dalej metody.

4 Metody i narzędzia reprezentacji reguł

Reguły w Semantic Web pojawiają się w miejscu, gdzie nie wystarczają ontologie, na przykład do reprezentowania klauzul Horna, a także do tworzenia silników wnioskujących.

Językiem zapisu reguł dla Semantic Web jest RIF - Rule Interchange Format. Język ten ma być w założeniu niezależny od poszczególnych systemów regułowych umożliwiając zapis reguł z dowolnych systemów. Idealny model integrujący wszystkie możliwe reprezentacje reguł w systemach regułowych jest niewykonalny - stąd pomysł, by zdefiniować rdzenie dla poszczególnych rodzin języków oraz ich warianty. Okazuje się jednak, że i to nie wystarcza na większą skalę. Pojawił się więc pomysł stworzenia hierarchii rdzeni:

- Basic Logic Dialect - Jest to ogólny język regułowy oparty na logice, niezależny od technologii Semantic Web. Umożliwia zapis pozytywnych klauzul Horna. Może być stosowany jako *język regułowy* lub *format wymiany danych*.
- Production Rule Dialect - kolejny rdzeń dla rodzin języków wykorzystujących reguły postaci *if-then*
- RIF Core służy do połączenia obu wyżej wymienionych.

Zarówno RIF-BLD jak i RIF-PRD są aktualnie w fazie rozwoju, a opisujące je dokumenty mają status Working Draft.

RuleML to język znaczników stworzony do opisu reguł wnioskowania w przód (bottom-up) oraz wstecz (top-down) w formacie XML. Umożliwia realizację dedukcji i przekształcanie reguł. Rozwijany jest przez Rule Markup Initiative, otwartą grupę składającą się z ośrodków akademickich i przemysłowych. Inicjatywa rozpoczęła się w roku 2000. RuleML oparty jest na Datalogu - języku będącym częścią wspólną SQL i Prologu. W Datalogu możemy definiować fakty odpowiadające wierszom relacyjnych baz danych oraz reguły odpowiadające tabelom zdefiniowanym przez widoki. W RuleML można zapisać n-elementowe relacje.

Dla reguły zapisanej w naturalnym języku angielskim: „*The discount for a customer buying a product is 7.5 percent if the customer is premium and the product is luxury.*” zapis w RuleML wygląda następująco:

```
<Implies>
  <head>
    <Atom>
      <Rel>discount</Rel>
      <Var>customer</Var>
      <Var>product</Var>
      <Ind>7.5 percent</Ind>
    </Atom>
  </head>
  <body>
    <And>
      <Atom>
        <Rel>premium</Rel>
        <Var>customer</Var>
      </Atom>
      <Atom>
        <Rel>luxury</Rel>
        <Var>product</Var>
      </Atom>
    </And>
  </body>
</Implies>
```

SWRL czyli Semantic Web Rule Language jest propozycją języka reguł łączącego podzbiory OWL (OWL DL i OWL Lite) z podzbioremi RuleML (Unary/Binary Datalog). Specyfikacja została przedłożona w 2004 roku przez National Research Council of Canada, Network Inference i Stanford University. SWRL jest złączeniem, w sensie sumy zbiorów, a nie ich przecięcia logiki opartej na klauzulach Horna i OWL. SWRL zachowuje moc OWL DL, kosztem braku rozstrzygalności. Reguł zapisywane są w postaci implikacji między przesłankami (body) a konkluzją (head). Kodowanie SWRL odbywa się za pomocą tzw. XML Concrete Syntax. Jest to połączenie *OWL Web Ontology Language XML Presentaion Syntax* oraz *RuleML XML Syntax*. Na przykład zapis reguły: $hasParent(?x1, ?x2) \wedge hasBrother(?x2, ?x3) \rightarrow hasUncle(?x1, ?x3)$ w składni XML Concrete Syntax wygląda następująco:

```
<ruleml:imp>
  <ruleml:_rlab ruleml:href="#example1"/>
  <ruleml:_body>
    <swrlx:individualPropertyAtom swrlx:property="hasParent">
      <ruleml:var>x1</ruleml:var>
      <ruleml:var>x2</ruleml:var>
    </swrlx:individualPropertyAtom>
    <swrlx:individualPropertyAtom swrlx:property="hasBrother">
```

```

        <ruleml:var>x2</ruleml:var>
        <ruleml:var>x3</ruleml:var>
    </swrlx:individualPropertyAtom>
</ruleml:_body>
<ruleml:_head>
    <swrlx:individualPropertyAtom swrlx:property="hasUncle">
        <ruleml:var>x1</ruleml:var>
        <ruleml:var>x3</ruleml:var>
    </swrlx:individualPropertyAtom>
</ruleml:_head>
</ruleml:imp>

```

Metody modelowania reguł w Sieci Semantycznej nie odnoszą się bezpośrednio do wizualnego projektowania bazy reguł, nie przewidują także jej strukturalizacji, czy hierarchizacji. Tego typu rozwiązania są natomiast dostępne w omówionym w następnej części projekcie HeKatE.

5 Regułowa reprezentacja wiedzy w projekcie HeKatE

Projekt HeKatE (hecate.ia.agh.edu.pl) jest realizowany w ramach grantu badawczego kierowanego przez Prof. Antoniego Ligęzę w Katedrze Automatyki AGH. Projekt związany jest z integracją wybranych metod inżynierii wiedzy [6,7] w inżynierii oprogramowania. Podstawowe cele projektu to:

- formalizacja składni i semantyki systemów regułowych,
- efektywna strukturalizacja bazy wiedzy, wraz z hierarchizacją procesu wnioskowania,
- wsparcie dla wizualnego projektowania systemu,
- integracja jądra logiki regułowej z aplikacjami w systemach biznesowych i sterowania,
- analiza jakości systemu pod kątem istotnych własności formalnych.

W ramach projektu rozważany jest cały proces projektowania systemu z bazą wiedzy. Proces projektowania w podejściu HeKatE obejmujący następujące fazy:

- **modelowanie konceptualne**, w którym identyfikuje się system atrybutów i ich związki funkcjonalne. W fazie tej korzysta się z metodologii modelowania ARD [9], która umożliwia określenie funkcjonalnych zależności między atrybutami systemu za pomocą ich graficznych reprezentacji.
- **projektowanie logiczne**, w którym struktura systemu jest reprezentowana jako struktura hierarchiczna XTT [10]. W fazie tej struktura systemu może być poddana analizie, weryfikacji, a także optymalizacji (w języku Prolog).
- **projektowanie fizyczne**, w którym dla modelu XTT generowany jest kod np. w Prologu, który może być skompilowany, wykonany i poprawiony [11].

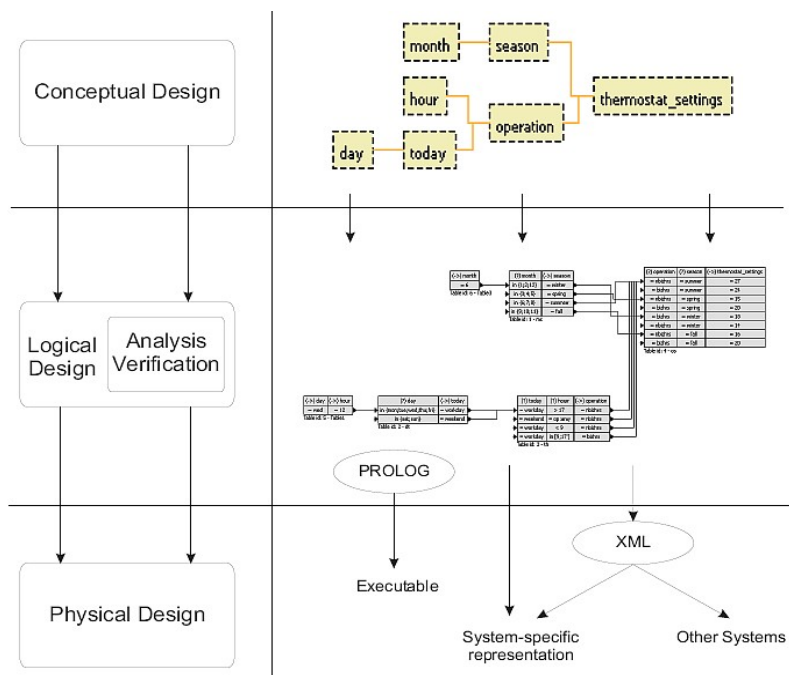
Na Rys. 2 przedstawiono cały proces projektowania XTT dla przykładowego problemu termostatu (system ustawiania temperatury dla biura). W procesie tym są używane odpowiednie metody reprezentacji wiedzy.

Metoda ARD ma za zadanie określić funkcyjne zależności między elementami (własnościami) systemu. Jej podstawowym celem jest wsparcie prototypowania reguł decyzyjnych opisanych przy pomocy reprezentacji XTT.

XTT jest formalizmem reprezentacji wiedzy regułowej pozwalającym na ustrukturyzowanie bazy reguł poprzez wprowadzenie tabel grupujących reguły posiadające te same atrybuty oraz łączy pomiędzy tabelami, pozwalających na sterowanie wnioskowaniem między tabelami.

W fazie projektu fizycznego prototyp systemu regułowego jest opisywany przy pomocy metareprezentacji, która jest następnie uruchamiana przez dedykowany interpreter [11].

Istotnym aspektem metodologii jest pełna formalizacja składni i semantyki reguł XTT. Jest ona zrealizowana przy pomocy logiki ALSV(FD) pozwalającej na rozszerzenie prostej reprezentacji atrybutowej o użycie wartości zbiorowych atrybutów, zapewniając przy tym metody efektywnego wnioskowania [12].



Rysunek 2. Fazy zintegrowanego procesu projektowania XTT w HeKatE

6 Propozycja integracji

Celem proponowanej integracji reprezentacji reguł jest z jednej strony dostarczenie efektywnych metod wizualnego projektowania i analizy reguł dla Sieci Semantycznej, z drugiej zaś dopasowanie generycznych metod HeKatE do aplikacji na tej platformie.

Rozważane są następujące poziomy integracji:

- logiki opisu, na etapie którego przeprowadzona jest analiza syntaktyczno-semantyczna Logiki Opisowych oraz Logiki Atrybutowej oraz zaproponowany jest model przejścia między tymi formalizmami,
- słownika reguł decyzyjnych, na którym opis stanu systemu w Logice Atrybutowej zestawiony jest z ontologią formalną w Description Logics
- pojedynczych reguł, operujących na pojęciach zdefiniowanych w ontologii będącej słownikiem dla reguł,
- struktury bazy wiedzy,
- wymiany reguł, za pomocą standardowych formatów oraz
- uruchamiania reguł.

Poniżej opisany jest każdy z nich.

Na poziomie logiki opisu trwają prace nad sformalizowaniem przejścia pomiędzy DL a ALSV(FD). W obu podejściach mówi się o pojęciach/atributach i ich wartościach/instancjach. Tym niemniej semantyka i zadania wnioskowania obu rachunków są inne. Celem prac jest umożliwienie opisu słownika dla reguł wyrażonego w ARD przy pomocy DL i ontologii.

W metodzie ARD modelowane są zależności funkcyjne pomiędzy atrybutami, oraz ich hierarchia w procesie przyrostowego projektowania systemu. Na tym etapie integracji planowany jest opis diagramów ARD przy pomocy języka OWL.

Kolejne poziomy obejmują modelowanie reguł i struktury bazy wiedzy. W tym pierwszym przypadku możliwe jest opisanie reguł XTT w oparciu o słownik zbudowany w OWL. Zatem XTT pasowałby się na poziomie SWRL. Jeżeli chodzi o strukturę bazy wiedzy, w Sieci Semantycznej w chwili obecnej nie są rozważane takie mechanizmy. Co za tym idzie wymiana takich reguł musiałaby być realizowana na poziomie RIF.

Uruchamianie reguł XTT byłoby w tym przypadku możliwe z użyciem mechanizmów dostępnych dla Sieci Semantycznej, wykorzystujących DL. Programy wnioskujące, tak zwane *reasonery*, wykonujące wnioskowanie na poziomie rachunku predykatów, posłużyłyby do sprawdzania warunków reguł decyzyjnych. Z racji tego, iż te rozwiązania są obecnie we wczesnej fazie rozwoju, należy się liczyć z koniecznością opracowania rozszerzonego interpretera.

7 Kryteria ewaluacji

Analizując zalety oraz wady proponowanego rozwiązania oraz porównując je z istniejącymi rozwiązaniami, warto zwrócić uwagę na dwa aspekty istnienia reguł w środowisku sieci Internet. Z jednej strony projekt Semantic Web doczekał się licznych rekomendacji i innych dokumentów konsorcjum W3C, w tym opracowań wspomnianych już języków regułowych RIF oraz SWRL. Z drugiej, w środowisku biznesowym popularność zyskały silniki regułowe takie jak Drools czy Jess, umożliwiające deklaratywne programowanie systemów ekspertowych uruchamianych w aplikacjach internetowych. Mamy więc do czynienia zarówno ze standardami opracowanymi teoretycznie z ograniczonymi przykładami oraz standardami *de facto* istniejącymi w działających aplikacjach.

Proponowane kryteria ewaluacji to:

- kompatybilność z formalizmem Description Logics, na którym oparte są ontologie mające fundamentalne znaczenie dla aplikacji Semantic Web,
- ekspresyjność oraz wydajność obliczeniowa,
- wymiana reguł pomiędzy systemami w oparciu o standardy RIF,
- środowisko uruchomieniowe dla reguł,

W tym kontekście proponowane rozwiązanie podejmuje wyzwania ujęte w powyższe kryteria ewaluacji.

Od strony formalnego opisu opracowany jest model przejścia pomiędzy Logiką Atrybutową oraz Logikami Opisowymi, na poziomie składni i semantyki. Celem tego przejścia jest możliwość wykorzystania przez reguły pojęć opisanych w ontologiach. Zapis w Logice Atrybutowej jest bardziej ograniczony i nie wszystkie konstrukcje dopuszczalne w Description Logics można zapisać za pomocą rachunku atrybutów. O ile DL w dużej mierze opiera się na rachunku predykatów I rzędu, rachunek atrybutowy rozszerza semantykę logiki rzędu 0 czyli rachunku zdań. Proponowane rozwiązanie nie ma jednak za zadanie pokryć całej funkcjonalności DL, lecz ją uzupełnić i wzbogacić w konkretnych przypadkach użycia.

Zalety proponowanego podejścia do integracji to między innymi dostarczenie efektywniejszych mechanizmów wnioskowania dla reguł w Sieci Semantycznej, oraz wsparcia dla wizualnego projektowania takich reguł. Używana w HeKatE logika ALSV(FD) jest z jednej strony dużo bardziej ekspresywna od rachunku zdań, na którym opierają się klasyczne systemy regułowe, z drugiej zaś efektywniejsza we wnioskowaniu od rachunku predykatów, w związku z brakiem zmiennych logicznych i ograniczeniom na poziomie semantyki do mechanizmów odpowiadających predykatom unarnym. Nie pozwala to wprawdzie na przeprowadzanie wnioskowania na takim poziomie jak w przypadku używanych w SWRL klauzul Horna, jest jednak wystarczające w przypadku prostych systemów decyzyjnych opartych na wnioskowaniu w przód. Rozwijane w HeKatE metody wizualnego projektowania i formalnego analizowania reguł, pozwalają na znaczące podniesienie jakości procesu projektowania takich systemów.

Na etapie uruchomienia mamy możliwość skorzystania z istniejących silników wnioskujących, takich jak Drools, czy Jess oraz generycznego narzędzia dedykowanego dla metodologii XTT rozwijanego w ramach projektu HeKatE. Z drugiej strony istniejące programy wnioskujące (ang. *reasoners*) dla języka OWL opartego na DL wykorzystują mechanizmy typowe dla rachunku pierwszego rzędu. Ich wnioskowanie na poziomie zależności pomiędzy pojęciami w ontologii może zostać wykorzystane do sprawdzenia warunków reguł decyzyjnych w danym momencie.

8 Przyszłe prace

W artykule omówione zostały najważniejsze aspekty architektury Sieci Semantycznej, wraz z metodami reprezentacji reguł. Obecnie jest to obszar najbardziej aktywnych prac. W tym kontekście, wyniki projektu HeKatE rozwijającego klasyczne metody systemów ekspertowych, pozwalające na wizualne projektowanie i strukturalizację bazy reguł mogą się okazać cennym uzupełnieniem.

Przyszłe badania nad integracją będą skupione nad formalizacją translacji z ALSV(FD) do DL oraz opisie słownika ARD za pomocą ontologii i języka OWL. Z drugiej strony trwają intensywne prace nad integracją narzędzi projektowych i formatów wymiany wiedzy regulowej używanych w projekcie HeKatE z rozwiązaniami wykorzystywanymi w Sieci Semantycznej.

Podziękowania

Praca naukowa finansowana ze środków na naukę w latach 2007-2009 jako projekt badawczy HeKatE.

Bibliografia

1. Berners-Lee T., Hendler J., Lassila O., (2001). *The Semantic Web*, May 17.
2. Berners-Lee T., (2000). *Semantic Web - XML*, <http://www.w3.org/2000/Talks/1206-xml2k-tbl/Overview.html>.
3. Herman I., (2007). *W3C - Introduction to the Semantic Web*, International Conference on Dublin Core and Metadata Applications, Singapore, 2007-08-31.
4. Dubost K., Herman I., (2008). *W3C - State of the Semantic Web*, INTAP Semantic Web Conference, Tokyo, Japan, March 7, 2008.
5. Liebowitz, J., (2000). *The Handbook of Applied Expert Systems*. CRC Press.
6. Russell, S., Norvig, P., (2003). *Artificial Intelligence: A Modern Approach*. 2nd edn. Prentice-Hall.
7. Harmelen, F. van, Lifschitz, V., Porter, B., (2007). *Handbook of Knowledge Representation*. Elsevier Science.
8. Negnevitsky, M., (2002), *Artificial Intelligence. A Guide to Intelligent Systems*. Addison-Wesley.
9. Ligeza, A., (2006), *Logical Foundations for Rule-Based Systems*. Berlin, Heidelberg: Springer-Verlag.
10. Nalepa, G. J., Ligeza, A., (2005). *A graphical tabular model for rule-based logic programming and verification*. *Systems Science*, 31(2):8995.
11. Nalepa, G. J., Ligeza, A., (2006). *Prolog-based analysis of tabular rule-based systems with the XTT approach*. In Geoffrey C. J. Sutcliffe and Randy G. Goebel, editors, FLAIRS 2006 : proceedings of the nineteenth in-ternational Florida Artificial Intelligence Research Society conference : [Melbourne Beach, Florida, May 1113, 2006], pages 426431, FLAIRS. - Menlo Park, 2006. Florida Artificial Intelligence Research Society, AAAI Press.
12. Nalepa G. J., Ligeza A., (2008), *Xtt+ rule design using the ALSV(FD)*. In Adrian Giurca, Anastasia Analyti, and Gerd Wagner, editors, ECAI 2008: 18th European Conference on Artificial Intelligence: 2nd East European Workshop on Rule-based applications, RuleApps2008: Patras, 22 July 2008, pages 1115, Patras, 2008. University of Patras.