

Politechnika Poznańska

Streszczenie rozprawy doktorskiej

**Regułowa metoda realizacji zapytań
do bazy wiedzy o przestępstwach
gospodarczych**

Jarosław Ksawery BAŁ

Promotor:

prof. dr hab. inż. Czesław JĘDRZEJEK

Praca przygotowana
w Instytucie Automatyki i Inżynierii Informatycznej
Politechniki Poznańskiej



Poznań, 2013

Spis treści

1	Wstęp	1
1.1	Motywacje	2
1.2	Cele pracy	4
1.3	Opublikowane wyniki	5
2	Podstawy teoretyczne	7
3	Baza wiedzy o przestępstwach gospodarczych	9
3.1	Sprawa Hydry	9
3.2	Model Ontologii Minimalnej	10
4	Metody regułowej realizacji zapytań	15
4.1	Metoda Hybrydowa	16
4.2	Metoda Rozszerzonych Reguł	17
4.3	Mapowanie Predykatów Ontologicznych na Dane Relacyjne	18
5	Implementacja biblioteki SDL	21
6	Eksperymenty obliczeniowe	23
7	Wnioski	25
	Bibliografia	27

Rozwój relacyjnych baz danych dostarczył wydajnych rozwiązań do gromadzenia, przetwarzania i zarządzania dużymi zbiorami danych. Pomimo wielu zalet systemów relacyjnych nie umożliwiają one przetwarzania danych zgodnie z zawartą w nich semantyką. Współczesny rozwój systemów wymaga nie tylko zarządzania danymi, ale również wiedzą, które ze sobą niosą. Ponadto, schemat relacyjnej bazy danych często jest znany i rozumiany tylko przez samych twórców. W wielu przypadkach nazwy tabel i kolumn nie zawsze odzwierciedlają przechowywane dane. Takiej rozbieżności nazw nie da się uniknąć bez jawnie zdefiniowanej semantyki opisującej znaczenie przechowywanych danych.

Ostatnie dziesięciolecie to dynamiczny rozwój mechanizmów służących definiowaniu znaczenia danych. Ontologie, bo o nich mowa, znajdują zastosowanie w technologiach Semantycznej Sieci WWW, integracji danych, systemach wieloagentowych, narzędziach przetwarzania języka naturalnego i wielu innych. Ontologie, jako jawnie zdefiniowana semantyka danych, stanowią element uzupełniający dla systemów relacyjnych baz danych. Integracja tego typu baz wraz z ontologiami umożliwiającą zadawanie zapytań wydaje się być interesującym i przyszłościowym rozwiązaniem.

Zadawanie zapytań do relacyjnych baz danych jest ograniczone do poziomu abstrakcji reprezentowanego przez schemat bazy. Tego typu zapytania nie dość, że nie oddają semantyki danych to ich zadawanie wymaga znajomości konstrukcji języka SQL (ang. *Structured Query Language*). Wprowadzenie ontologii do systemów relacyjnych baz danych pozwala na formułowanie zapytań na poziomie semantycznym i jest dużo prostsze niż w przypadku języka SQL. Transformując ontologię w zbiór reguł wnioskujących i wykorzystując silnik wnioskujący (ang. *reasoning engine*) do udzielenia odpowiedzi na zapytanie, mamy do czynienia z metodą regułowej realizacji zapytań.

W rozprawie doktorskiej podejmowany jest problem regułowej metody realizacji zapytań (RRZ) wraz z jej praktycznym zastosowaniem do bazy wiedzy o przestępstwach gospodarczych. Takie zastosowanie metod RRZ pozwoli na odkrywanie działań przestępczych jak również na proponowanie kwalifikacji prawnej popełnionych czynów dla osób biorących udział w tych działaniach.

1.1 Motywacje

Systemy regułowe, będące jedną z metod sztucznej inteligencji, bazują na dobrze zdefiniowanej, symbolicznej reprezentacji wiedzy wyrażonej w postaci reguł. Ogólna postać reguł prezentuje się następująco: jeżeli przesłanki są spełnione to dodaj konkluzje w postaci faktów do bazy wiedzy (przesłanki \rightarrow konkluzje). Baza wiedzy w systemach regułowych składa się z dwóch części: zbioru reguł (część intensjonalna) oraz zbioru faktów (część ekstensjonalna), czyli danych.

Problem regułowej realizacji zapytań był przedmiotem wielu badań [Bry 2007]. Jest on bezpośrednio związany z procesem wnioskowania biorącym udział w ewaluacji zapytania. Silnik wnioskujący przetwarza reguły zgodnie z algorytmem wnioskowania. W zależności od zastosowanej metody (wstecz lub w przód) przetwarzanie reguł sterowane jest celem (zapytaniem) bądź danymi (w przód). W rezultacie otrzymujemy odpowiedź na zadane zapytanie. Wyniki inicjatywy OpenRuleBench [Liang 2009], mającej na celu porównanie efektywności silników wnioskujących, pokazują że narzędzia takie jak Prolog, czy techniki dedukcyjnych baz danych wypadają dużo lepiej niż systemy wnioskujące w przód. Ponadto, większość systemów wnioskujących wymaga danych w pamięci roboczej, która jest ograniczona dostępną pamięcią operacyjną RAM. Takie ograniczenie stanowi poważną barierę dla skalowalności systemów regułowych.

Coraz szersze zastosowanie tego typu systemów wymaga opracowania nowych usprawnień umożliwiających efektywniejsze wnioskowanie oraz zadawanie zapytań. Dotychczasowe rozwiązania systemów regułowych nie radzą sobie z efektywnym przetwarzaniem dużych zbiorów danych w procesie realizacji zapytań. Zwiększenie skalowalności oraz usprawnienie obsługi zapytań w systemach regułowych jest elementem motywującym do prowadzenia prac badawczych z zakresu sztucznej inteligencji.

Dodatkowym czynnikiem motywującym do prowadzenia badań naukowych jest fakt, że coraz bardziej powszechny dostęp do nowych technologii zmienia sposób myślenia, pracy oraz działania w instytucjach publicznych (i nie tylko). Dlatego zasadnym wydaje się być proponowanie rozwiązań oraz narzędzi wspomagających pracę osób zajmujących się wykrywaniem przestępstw gospodarczych (prokuratorzy, śledczy), które często mają do czynienia z dużymi zbiorami danych, przekraczającymi możliwości analityczne człowieka.

W celu przyspieszenia oraz wsparcia procesów wykrywania przestępstw należałoby wykorzystać zaawansowane technologicznie narzędzia z wyspecyfikowaną wiedzą prawną. Do takich narzędzi należą między innymi systemy regułowe wykorzystujące automatyczne wnioskowanie wraz z mechanizmami obsługi zapytań, jak również ontologie jawnie definiujące semantykę danych. Zapytania kierowane do takiego systemu mogą potwierdzić bądź odrzucić hipotezy stawiane przez osoby zaangażowane w wykrywanie przestępstw gospodarczych. W ten sposób proces ten

ulegnie znacznym usprawnieniom oraz będzie mógł być przeprowadzany szybciej niż w tradycyjnym, „papierowym”, podejściu.

Problem stosowania systemów regułowych w dziedzinie prawa nie jest zagadnieniem nowym. Ze względu na jego złożoność proponuje się wiele różnych rozwiązań, które nie zawsze uwzględniają realne potrzeby. Badania prowadzone w ramach prac Polskiej Platformy Bezpieczeństwa Wewnętrznego (PPBW), w których autor rozprawy uczestniczył, wykazały między innymi, że:

- 60% przestępstw gospodarczych dotyczy działań na szkodę spółki w postaci przestępczego rozporządzania majątkiem,
- 90% wiedzy o takich przestępstwach ma postać faktów dotyczących transakcji bankowych, bądź dokumentów w obiegu gospodarczym,
- formalizacja dziedziny prawa dotyczącej przestępstw gospodarczych nie została dostatecznie zbadana i wymaga usprawnień.

Ograniczenia systemów regułowych w zakresie realizacji zapytań jak również mechanizmów wnioskowania są wystarczającym bodźcem do prowadzenia badań naukowych. Możliwość praktycznego zastosowania uzyskanych wyników do wspomaganie wykrywania i analizy przestępstw gospodarczych utwierdza w przekonaniu o słuszności podjętych badań.

Korzyści płynące z opracowanej metody RRZ mogą być następujące:

- Nowa metoda integracji ontologii, reguł, silników wnioskujących oraz relacyjnych baz danych.
- Nowa metoda realizacji zapytań konstruowanych z predykatów ontologicznych, co przedkłada się na prostszy sposób ich definiowania niż w przypadku konstrukcji języka SQL. W rezultacie, zapytania mogłyby być definiowane przez użytkowników takich jak prokuratorzy, śledczy czy policjanci.
- Zwiększenie efektywności działania silnika wnioskującego w kontekście przetwarzania reguł oraz realizacji zapytań. Wykorzystany silnik wnioskujący Jess [Hill 2003] implementuje algorytm Rete [Forgy 1982] i jest uważana za wiodące narzędzie służące do wnioskowania.
- Możliwość zastosowania opracowanych metod w innych narzędziach wnioskujących wykorzystujących algorytm Rete, np. Drools 5 [Community 2012].
- Usprawnienie pracy prokuratorów i śledczych w analizie przestępstw gospodarczych oraz ułatwienie przyporządkowywania kwalifikacji prawnej czynów do osób biorących czynny udział w przestępstwach gospodarczych.

- Możliwość zastosowania opracowanych metod w innych dziedzinach, co wymagałoby jedynie opracowania odpowiednich baz wiedzy. Dodatkowo, metody RRZ znalazłyby zastosowanie w aplikacjach, gdzie wymagana jest prostota definiowania zapytań, jak również semantyczny opis danych.

1.2 Cele pracy

Mając na uwadze ograniczenia systemów regułowych, możliwości składowania danych w bazach relacyjnych, metody wyrażania semantyki, a także kontekst praktycznego zastosowania, określono następujący cel pracy:

Zaprojektowanie metody regułowej realizacji zapytań z wykorzystaniem relacyjnej bazy danych oraz formalnie zdefiniowanej semantyki.

Założono również, że opracowana metoda zostanie użyta razem z bazą wiedzy o przestępstwach gospodarczych opisującej dwa rodzaje przestępstw: sprzeniewierzenie pieniędzy (malwersacja) i pranie pieniędzy. Powstały w ten sposób system pozwalałaby na wykrywanie działań o charakterze przestępczym jak również sugerowałby możliwe kwalifikacje prawne popełnionych czynów. Baza wiedzy reprezentowana w logice deskrypcyjnej wraz z regułami ma za zadanie odzwierciedlać hierarchię pracowników (właściciel, manager, dyrektor) wraz z ich poziomem odpowiedzialności w firmie, dokumenty wraz z ich atrybutami (faktury, daty, towary) a także parametry poszczególnych transakcji (przelew, wypłata, kwota). Ponadto, baza wiedzy zawierać musi informacje o artykułach polskiego kodeksu karnego [Sejm 1997] wraz z ich regułową definicją (gdzie przesłanki określają warunki możliwe do ustanowienia kwalifikacji prawnej). Baza wiedzy o przestępstwach gospodarczych wyrażona w formie ontologii musi mieć możliwość reprezentacji w formie reguł, ze względu na zastosowanie metody RRZ.

Opracowana metoda regułowej realizacji zapytań powinna mieć zastosowanie ogólne. Oznacza to, że jej użycie wymagałoby tylko wykorzystania innej bazy wiedzy (ontologii dziedzinowej).

Osiągnięcie głównego celu pracy wymaga realizacji następujących zadań:

- Identyfikacja odpowiedniego formalizmu, który pozwala na połączenie logiki deskrypcyjnej wraz z regułami (dowolna kombinacja prowadzi do nierozstrzygalności procesów wnioskowania).
- Określenie algorytmu wnioskującego, który będzie przetwarzał reguły. Ze względu na aplikacyjny charakter prac, zadanie to wymaga wybrania silnika wnioskującego działającego zgodnie z wybranym algorytmem.

- Zaprojektowanie metod RRZ, które okażą się znacznym usprawnieniem aktualnie istniejących systemów wnoszących zachowując warunek rozstrzygalności procesów wnioskowania.
- Opracowanie metody mapowania pomiędzy predykatami ontologicznymi a danymi relacyjnymi. Efektem tej metody będzie integracja ontologii, reguł oraz relacyjnej bazy danych.
- Skonstruowanie bazy wiedzy o przestępstwach gospodarczych: sprzeniewierzeniu oraz praniu pieniędzy. Baza wiedzy będzie wyrażona w formie ontologii w logice deskrypcyjnej wraz z regułami.

1.3 Opublikowane wyniki

Większość elementów rozprawy została opublikowana w wydawnictwach specjalistycznych jak również czasopismach międzynarodowych. Lista publikacji, wraz z numerami rozdziałów których dotyczą, jest następująca:

Rozdział 3:

- Bąk J., Jędrzejek C., *Application of an ontology-based model to a selected fraudulent disbursement economic crime*. In Proceedings of the 2009 international conference on AI approaches to the complexity of legal systems: complex systems, the semantic web, ontologies, argumentation, and dialogue, AICOL-I/IVR-XXIV'09, 113–132, Berlin, Heidelberg, 2010. Springer-Verlag.
- Bąk J., Jędrzejek C., Falkowski M., *Application of an ontology-based and rule-based model to selected economic crimes: fraudulent disbursement and money laundering*. In Proceedings of the 2010 international conference on Semantic web rules, RuleML'10, 210–224, Berlin, Heidelberg, 2010. Springer-Verlag.
- Jędrzejek C., Cybulka J., Bąk J., *Towards ontology of fraudulent disbursement*. In Proceedings of the 5th KES international conference on Agent and multi-agent systems: technologies and applications, KES-AMSTA'11, pages 301–310, Berlin, Heidelberg, 2011, Springer-Verlag.
- Bąk J., Cybulka J., Jędrzejek C., *Ontological Modeling of a Class of Linked Economic Crimes*. T. Computational Collective Intelligence, vol. 9, 98–123, 2013.

Rozdział 4:

- Bąk J., Jędrzejek C., *Querying relational databases using ontology, rules and Jess reasoning engine*. Studia z Automatyki i Informatyki, vol. T. 33, 24–44, 2008.
- Bąk J., Jędrzejek C., Falkowski M., *Usage of the Jess Engine, Rules and Ontology to Query a Relational Database*. In Proceedings of the 2009 International Symposium on Rule Interchange and Applications, RuleML '09, 216–230, Berlin, Heidelberg, 2009. Springer-Verlag.
- Bąk J., Brzykcy G., Jędrzejek C., *Extended rules in knowledge-based data access*. In Proceedings of the 5th international conference on Rule-based modeling and computing on the semantic web, RuleML'11, 112–127, Berlin, Heidelberg, 2011. Springer-Verlag.

Rozdziały 5 i 6:

- Bąk J., Falkowski M., Jędrzejek C., *Application of the SDL Library to Reveal Legal Sanctions for Crime Perpetrators in Selected Economic Crimes: Fraudulent Disbursement and Money Laundering*. In Monica Palmirani, M. Omair Shafiq, Enrico Francesconi and Fabio Vitali, editors, Proceedings of the RuleML-2010 Challenge, at the 4th International Web Rule Symposium, Washington, DC, USA, October, 21-23, 2010, volume 649 of CEUR Workshop Proceedings. CEUR-WS.org, 2010.
- Bąk J., Falkowski M., Jędrzejek C., *The SDL Library: Querying a Relational Database with an Ontology, Rules and the Jess Engine*. In Stefano Bragaglia, Carlos Viegas Damásio, Marco Montali, Alun Preece, Charles Petrie, Mark Proctor and Umberto Straccia, editors, Proceedings of the 5th International RuleML2011@BRF Challenge, co-located with the 5th International Rule Symposium, Fort Lauderdale, Florida, USA, November 3-5, 2011, volume 799 of CEUR Workshop Proceedings. CEUR-WS.org, 2011.

W kolejnych rozdziałach niniejszego streszczenia omówiono pokrótce zawartość i główne osiągnięcia rozprawy doktorskiej.

Podstawy teoretyczne

W rozdziale drugim przedstawiono podstawowe pojęcia i definicje związane z regułową realizacją zapytań oraz przestępstwami gospodarczymi. Rozdział 2.1 swoim zakresem obejmuje:

1. Logikę pierwszego rzędu - jej składnię oraz semantykę.
2. Język Datalog [Gallaire 1978] służący do wyrażania reguł. Zaprezentowano składnię języka oraz jego semantykę. Ponadto, przedstawiono ewaluację zapytań wykorzystując metody wnioskowania wstecz oraz w przód.
3. Prezentację systemów regułowych na przykładzie silnika wnioskującego Jess [Hill 2003]. Zaprezentowano algorytm wnioskujący Rete a także jego implementację we wspomnianym silniku.
4. Metody optymalizacji ewaluacji zapytań w systemach: regułowych oraz bazujących na Datalogu. Opisano metody związane z transformacją *magic* [Bancilhon 1986], przekazywaniem wartości pomiędzy zmiennymi w regułach (sideways information passing strategy [Sagiv 1984]), jak również inne techniki optymalizacji.
5. Przedstawienie rodziny logik deskrypcyjnych [Baader 2003] wraz z ich składnią i semantyką. Poruszono najważniejsze zadania związane z wnioskowaniem przy użyciu logik deskrypcyjnych (opisowych).
6. Prezentację metod połączenia logiki deskrypcyjnej z regułami oraz ich wpływ na rozstrzygalność obliczeń. Przedstawiono takie metody jak: Semantic Web Rule Language [Horrocks 2004a, Horrocks 2005], Description Logic Programs [Grosz 2003], DL-safe Rules [Motik 2004], Description Logic Rules [Krötzsch 2008], Horn-*SHIQ* [Hustadt 2005] i inne. Zwrócono szczególną uwagę na formalizm Horn-*SHIQ*, w którym zaimplementowana została baza wiedzy o przestępstwach gospodarczych.

Rozdział 2.2 obejmuje opis dwóch przestępstw gospodarczych podejmowanych w pracy, mianowicie sprzeniewierzenia pieniędzy oraz prania pieniędzy. Sprzeniewierzenie pieniędzy polega na przywłaszczeniu sobie majątku, którego właścicielem się nie jest. Tego typu przestępstwa mają często miejsce w firmach, w których pracownicy przejmują pieniądze, które im się nie należą [Albrecht 2011], np. poprzez wystawienie faktury na fikcyjne towary z podaniem własnego konta celem wykonania przelewu. Pranie pieniędzy natomiast polega na wprowadzeniu

do legalnego obrotu gospodarczego pieniędzy pochodzących z nielegalnego procederu (źródła) [Salinger 2004]. Pranie pieniędzy jest zazwyczaj połączone z tzw. przestępstwem bazowym, w wyniku którego powstał nielegalny dochód, np. fundusze pochodzące ze sprzeniewierzenia pieniędzy, czy też ze sprzedaży narkotyków.

W rozprawie przedstawiono model sprzeniewierzenia pieniędzy wraz z praniem pieniędzy bazując na rzeczywistym przypadku tzw. *sprawy Hydry*.

Oba przedstawione przestępstwa gospodarcze mają ogromny, destrukcyjny wpływ na gospodarkę kraju a nawet Wspólnoty Europejskiej. W rezultacie problem ten pośrednio dotyka nie tylko sferę biznesową, ale również zwykłych obywateli. Tym bardziej zasadnym wydaje się być opracowanie metod wspomagających śledczych i prokuratorów w procesie analizy tego typu przestępstw wykorzystując technologie semantyczne wraz z regułą realizacją zapytań.

Baza wiedzy o przestępstwach gospodarczych

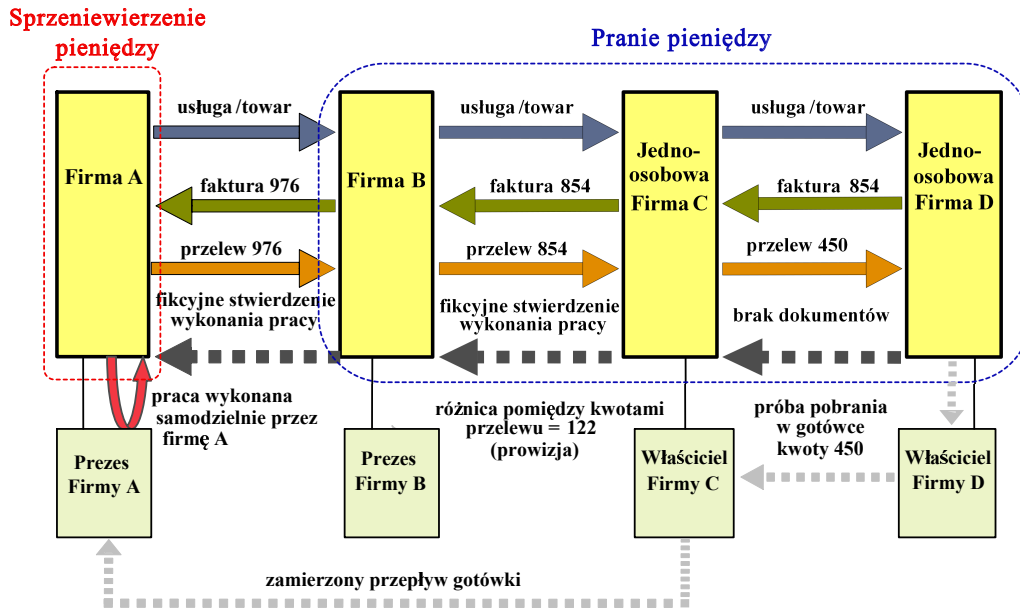
Rozdział zawiera opis oraz charakterystykę bazy wiedzy o dwóch, połączonych ze sobą, przestępstwach gospodarczych: sprzeniewierzeniu oraz praniu pieniędzy. Baza wiedzy powstała na podstawie analizy 10 prawdziwych przestępstw. Jako przykład modelowy wybrano tzw. *Sprawę Hydry*, w której popełniono oba typy przestępstw. Baza wiedzy nosi nazwę modelu ontologii minimalnej (MinOn) ze względu na fakt, że w jej konstruowaniu ograniczono się tylko i wyłącznie do najbardziej istotnych pojęć i relacji służących do opisu wspomnianych przestępstw.

W rozprawie zostały opisane najważniejsze elementy opracowanej ontologii z regułami: część terminologii dziedzinowej (wyrażonej w języku Horn-*SHIQ*) oraz części zadaniowej (wyrażonej w klauzulach Horna). Część terminologiczna ontologii (OT) zawiera pojęcia oraz relacje opisujące dziedzinę. Część zadaniowa ontologii (OZ), wykorzystuje składowe OT w regułach.

3.1 Sprawa Hydry

Sprawa Hydry to jeden z najprostszych przypadków sprzeniewierzenia oraz prania pieniędzy. Powstała baza wiedzy w pełni wyczerpuje opis tej sprawy wraz z jej możliwymi wariantami. Schemat podwójnego przestępstwa został przedstawiony na Rysunku 3.1.

Opis przestępstwa w oryginale zawierał 7 stron tekstu przygotowanego przez prokuratora. Opis ten następnie został przekształcony przez inżyniera wiedzy [Martinek 2008] do zdań, które w prosty sposób dało się zapisać jako trójki RDF przydatne do tworzenia bazy wiedzy. Wspomniany opis posłużył również do stworzenia schematu przedstawionego na Rysunku 3.1. Skrócony opis sprawy Hydry jest następujący. Prezes firmy *A* zlecił wykonanie prac budowlanych firmie *B*. Z kolei firma *B* zleciła te prace firmie *C*, a ta firmie *D*. Całość prac budowlanych wykonała wyłącznie firma *A*. Dokumenty stwierdzające wykonanie pracy oraz faktury zostały przesłane między firmami pozorując, jakoby firmy *B*, *C* i *D* wykonały zleczone prace. Następnie dokonano przelewów pieniężnych pomiędzy kontami bankowymi należącymi do firm biorących udział w schemacie. Firma *B* pobrała prowizję w wysokości 122 tysięcy złotych. Transfer pieniędzy z firmy *C* do *D* został



Rysunek 3.1: Sprawa Hydry [Martinek 2008]. Kwoty wyrażone w tysiącach złotych.

podzielony na dwie części. Jednakże do drugiego przelewu nie doszło, ze względu na fakt, że właściciel firmy *D* na drugi dzień po przelewie próbował wypłacić całość pieniędzy w gotówce. Jego zachowanie zwróciło uwagę pracowników banku i sprawa została zgłoszona do Generalnego Inspektora Informacji Finansowej, który wystosował pismo o możliwości popełnienia przestępstwa. W rezultacie w firmie *A* doszło do sprzeniewierzenia pieniędzy, natomiast w firmach *B*, *C* i *D* do prania pieniędzy. Jak się później okazało, część pieniędzy miała trafić do właściciela firmy *C* oraz prezesa firmy *A* (oznaczone na rysunku jako „zamierzony przepływ gotówki”).

Warto zaznaczyć, że przestępstwa typu sprzeniewierzenia (nie tylko pieniędzy) w 2009 roku stanowiły 2/3 wszystkich przestępstw gospodarczych na świecie [PricewaterhouseCoopers 2009]. Stanowi to kolejne uzasadnienie dla podjętych prac badawczych.

3.2 Model Ontologii Minimalnej

Opracowany model ontologii minimalnej (MinOn) opisującej dwa przestępstwa gospodarcze, został zbudowany zgodnie z metodologią inżynierską NeOn [Gómez-Pérez 2008]. Ontologia MinOn składa się z dwóch elementów: opisu dziedzinowego (OT) oraz części zadaniowej (OZ). Obie składowe dają się przetransformować do zbioru reguł w postaci klauzul Horna. Powstała w ten sposób baza

wiedzy może zostać wykorzystana w opracowanej metodzie regułowej realizacji zapytań.

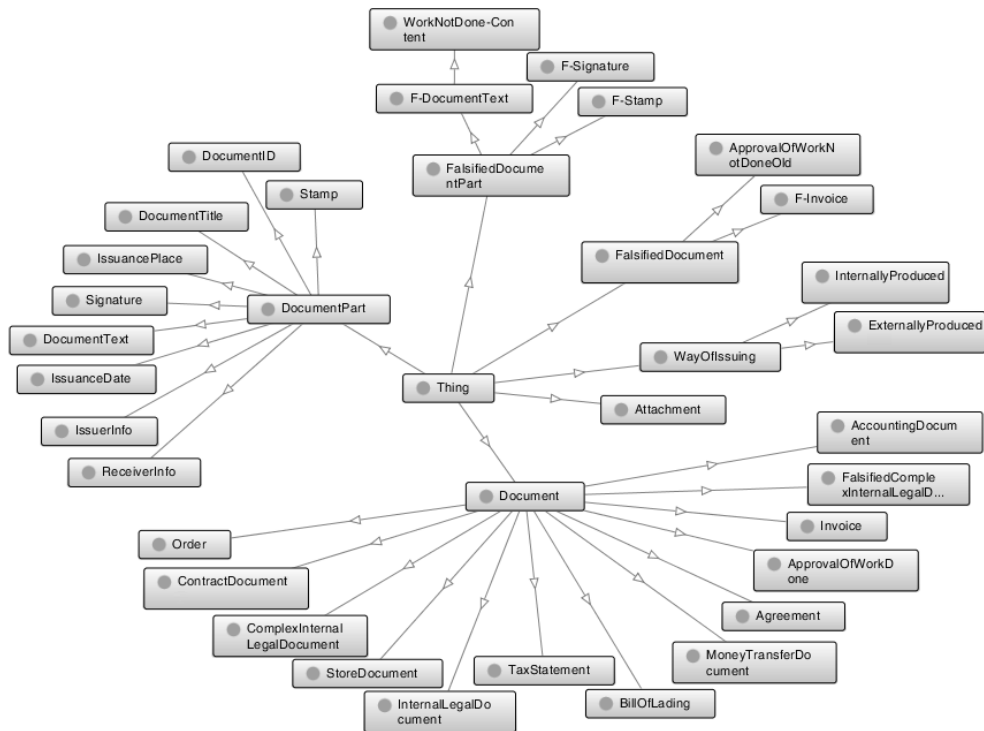
W ramach prac związanych z budową bazy wiedzy skupiono się na:

- Opisie faktów mających znaczenie w wykrywaniu działań przestępczych oraz sugerowaniu kwalifikacji prawnej czynów.
- Odseparowaniu trudnych kwestii i pozostawieniu ich człowiekowi, np. system nie może stwierdzić czy zlecona praca została wykonana tylko na podstawie dokumentów, ze względu na fakt, że mogą one być sfałszowane. Takie stwierdzenie wymaga wizji lokalnej i ustalenia czy np. firma *B* dysponowała sprzętem do wykonania prac budowlanych.
- Definiowaniu określonych atrybutów do minimum, np. w sprawie Hydry nie jest istotne, gdzie miały mieć miejsce prace budowlane – miasto, adres itd., chociaż takie informacje muszą się pojawić w akcie oskarżenia.
- Odzwierciedleniu w bazie wiedzy faktów istotnych z punktu widzenia działań biznesowych – przelewów, hierarchii osób w firmie, dokumentów itd. Są to „twarde” dowody, które są potwierdzone fizyczną reprezentacją (w przypadku przelewów są to dokumenty wydrukowane z systemu bankowego).

Baza wiedzy MinOn ma charakter modułowy i składa się następujących elementów:

- Person.owl – opisuje osoby, ich relacje a także grupy osób,
- Document.owl – opisuje znaczenie dokumentów oraz ich składowe (np. podpisy)
- LegalProvision.owl – definiuje akty prawne,
- Action.owl – opisuje podejmowane działania (akcje),
- Object.owl – opisuje inne elementy, np. towary, zlecenia itd.,
- MinimalModel.owl – definiuje ogólne pojęcia i relacje, zawiera ponadto reguły służące do wnioskowania,
- Institution-Organization.owl – opisuje firmy, pozycje w firmach oraz dopuszczalną w modelu ich strukturę (aktualnie rozważana jest trój-poziomowa hierarchia: Prezes, Dyrektor, Manager).

Przykład taksonomii w module Document.owl został zaprezentowany na Rysunku 3.2. Zbiór reguł zastosowanych w bazie wiedzy służy do wyrażenia zależności po-



Rysunek 3.2: Moduł Document.owl – graficzna reprezentacja taksonomii.

między osobami, dokumentami, przelewami bankowymi i kwalifikacjami prawnymi. Reguły zostały zaimplementowane w języku SWRL [Horrocks 2004b], przy czym forma reguł została ograniczona do klauzul Horna. Używając regułowej reprezentacji bazy wiedzy, użytkownik metody RRZ może kierować zapytania, które pozwolą mu na odkrycie nowych zależności oraz działań wskazujących na możliwość popełnienia przestępstwa. Ponadto, może otrzymać propozycje kwalifikacji prawnej czynów dla osób, które je popełniły (co również zostanie wykryte).

Reguły zdefiniowane w części zadaniowej bazy wiedzy wspierają wnioskowanie na temat:

- dokumentów i ich właściwości,
- hierarchii odpowiedzialności w firmie (im wyższe stanowisko tym większa),
- wykonanych transakcji (transakcja jako wirtualny byt powiązany z fakturą, przelewem i innymi dokumentami),
- akcji podejmowanych przez osoby,
- kwalifikacji prawnej czynów.

Skonstruowana baza wiedzy o przestępstwach gospodarczych zawiera 92 pojęcia, 60 ról, 3 właściwości typu OWLDatatype oraz 42 reguły (75 klauzul Horna, gdyż niektóre reguły mają koniunkcję predykatów w głowach). Baza wiedzy w pełni obejmuje sprawę Hydry oraz dopuszczalne warianty sekwencji wykonywanych czynności: kto dokładnie zlecił wykonanie prac, kto podpisał jakie dokumenty, kto miał uprawnienia do podpisywania, a także kto mógł nie wiedzieć o tym co podpisuje (w tym przypadku mamy do czynienia z zaniedbaniem lub niewinnością).

Metody regułowej realizacji zapytań

W niniejszym rozdziale przedstawiono dwie metody regułowej realizacji zapytań wykorzystujące bazę wiedzy w formie ontologii z regułami, relacyjną bazę danych oraz silnik wnioskujący.

Pierwsza z metod, metoda wnioskowania hybrydowego jest bezpośrednio zależna od wybranego silnika wnioskującego Jess [Hill 2003] oraz zaimplementowanego w niej algorytmu wnioskującego Rete [Forgy 1982]. Metoda tzw. rozszerzonych reguł jest efektywniejsza od metody hybrydowej i bardziej ogólna. Zależy ona tylko od algorytmu Rete i może z powodzeniem być zaimplementowana w innych silnikach wnioskujących niż Jess. Ponadto, ze względu na formę zapytań (predykaty ontologiczne) obie metody w znaczący sposób upraszczają konstrukcje zapytań w porównaniu do języka SQL.

W obu metodach przyjęto następujące założenia:

- Baza wiedzy wyrażona jest w logice deskrypcyjnej Horn-*SHIQ* wraz z regułami w postaci klauzul Horna. W regułach występują tylko i wyłącznie predykaty unarne, binarne oraz dodatkowe predykaty służące porównaniom ($<$, \leq , \neq , itd.).
- Baza wiedzy jest transformowana do postaci klauzul Horna. W rezultacie ontologia ma postać zbioru reguł.
- Zakłada się, że dopuszczalna forma zapytań obejmuje koniunkcję predykatów pochodzących z ontologii wraz z predykatami służącymi do porównań.
- Rozstrzygalność obliczeń jest zapewniona poprzez zastosowanie tzw. reguł DL-bezpiecznych [Motik 2004] oraz *Datalog safety* [Abiteboul 1995].
- W odniesieniu do semantyki zastosowane jest *założenie o zamkniętości świata*, które oznacza, że brak informacji jest informacją negatywną. Wynika to z faktu, że bazy danych (w których przechowujemy dane) oraz systemy regułowe również stosują to założenie. Ponadto, w prezentowanych metodach nacisk położony jest na wnioskowanie i realizację zapytań na „twardych” dowodach w sprawie (dochodzeniu).
- Jako algorytm wnioskowania wybrano algorytm Rete służący do wnioskowania w przód. Wybór podyktowany jest szerokim użyciem tego algorytmu w systemach akademickich oraz komercyjnych, i w konsekwencji, większego potencjalnego zastosowania opracowanych metod.

- Silnik wnioskujący, implementujący algorytm Rete, stanowi narzędzie Jess. Narzędzie to jest produktem darmowym do zastosowań naukowych oraz płatnym w przypadku innych zastosowań. Jess dostarcza mechanizmów wnioskowania w przód oraz wstecz (jest to de facto symulacja wnioskowania wstecz przy użyciu wnioskowania w przód).
- Fakty w silniku wnioskującym reprezentowane są w postaci trójek RDF [Beckett 2004, Horridge 2006].
- Dane trzymane są w relacyjnej bazie i mają postać tabel o ustalonej liczbie kolumn oraz wierszy. Dane te muszą być zamieniane „w locie” na trójki RDF w trakcie regułowej realizacji zapytań.
- Opracowane metody zostały zaimplementowane w narzędziu SDL (ang. *Semantic Data Library*). Językiem implementacji jest Java. Opis narzędzia został przedstawiony w rozdziale 5.
- Wszystkie transformacje reguł stosowane w obu metodach RRZ są wykonywane automatycznie poprzez narzędzie SDL.

4.1 Metoda Hybrydowa

Hybrydowa metoda regułowej realizacji zapytań wykorzystuje dwa mechanizmy wnioskowania: w przód oraz wstecz. Wykorzystywane są dwie instancje silnika wnioskującego Jess. Metoda jest bezpośrednio zależna od narzędzia wnioskującego Jess ze względu na wykorzystanie mechanizmu wnioskowania wstecz w nim zaimplementowanego.

Metoda hybrydowa składa się z następujących elementów:

- Zbiór faktów.
- Zbiór reguł do wnioskowania w przód (reguły są potrzebne tylko do generacji reguł wnioskujących wstecz).
- Zbiór reguł do wnioskowania wstecz.
- Zbiór reguł mapujących predykaty ontologiczne na dane relacyjne.
- Algorytm realizacji zapytań.

Zbiór faktów przechowywany jest w relacyjnej bazie danych, która jest odpytana zapytaniami SQL w trakcie wnioskowania. Dane, jako trójki, są pobierane z bazy za pomocą reguł mapujących, które zawierają zapytania SQL. W rezultacie dane są reprezentowane jako trójki (fakty) w silniku wnioskującym. Reguły mapujące wraz z regułami do wnioskowania wstecz znajdują się w silniku wnioskującym wstecz (WW). W silniku wnioskującym w przód (WP) znajduje się zapytanie oraz fakty przekopiowane z silnika WW. Taki podział zadań wynika z implementacji narzędzia Jess, gdzie porównywanie wartości pomiędzy zmiennymi jest wykonywane efektywniej podczas wnioskowania w przód. Zaproponowany algorytm

realizacji zapytań steruje dwoma silnikami wnioskującymi wymieniając fakty pomiędzy nimi, które powstają w wyniku procesu wnioskowania lub pochodzą z relacyjnej bazy danych. W wyniku działania algorytmu hybrydowego otrzymujemy odpowiedź na zadane zapytanie w postaci wartościowania zmiennych występujących w zapytaniu.

4.2 Metoda Rozszerzonych Reguł

Metoda rozszerzonych reguł (RR) stanowi modyfikację transformacji magic [Beeri 1987] zastosowaną z użyciem algorytmu Rete. Oryginalna transformacja magic jest bezpośrednio związana z porządkiem predykatów w regułach w zależności od zadanego zapytania. Warto tutaj nadmienić, że w zależności od zapytania (które zmienne są ukonkretnione, a które nie) można zbudować reguły różniące się porządkiem predykatów. Przy czym semantyka tych reguł pozostaje bez zmian. Ten istotny element transformacji magic został wykorzystany w prezentowanym podejściu.

Idea metody rozszerzony reguł bazuje na analizie zależności pomiędzy predykatami występującymi w regułach. Wprowadzono pojęcie tzw. *predykatów zależnych* definiujące zbiór predykatów, które współdzielą zmienną lub stałą. Zmienne występujące w predykatkach pozwalają na przekazywanie wartości pomiędzy nimi (np. współdzielą dokładnie tę samą zmienną lub wartość). Metoda rozszerzonych reguł składa się z następujących elementów:

- Zbiór faktów z podziałem na fakty *potrzebne* (ang. *called*) oraz na fakty *właściwe* (ang. *proper*).
- Zbiór reguł do wnioskowania w przód (reguły są potrzebne tylko do generacji reguł rozszerzonych).
- Zbiór rozszerzonych reguł (również do wnioskowania w przód).
- Zbiór reguł mapujących predykaty ontologiczne na dane relacyjne.
- Algorytm realizacji zapytań.

Podobnie jak w przypadku hybrydowej metody RRZ dane są trzymane w relacyjnej bazie danych i w wyniku zapytań SQL transformowane w formie trójek RDF do przestrzeni roboczej narzędzia Jess. Wykorzystywana jest jedna instancja narzędzia, która działa zgodnie z mechanizmem wnioskowania w przód. Podział na fakty właściwe i potrzebne jest istotny z punktu widzenia algorytmu realizacji zapytań. Fakty właściwe odpowiadają danym w relacyjnej bazie lub powstają w wyniku działania wnioskowania. Fakty potrzebne to cele, które są potrzebne, aby

udzielić odpowiedzi na zapytanie. Pozwalają one uniknąć uruchamiania nadmiarowych reguł, które nie są powiązane w żaden sposób z zapytaniem. Rozszerzone reguły są generowane automatycznie na bazie reguł definiujących bazę wiedzy (reguły podstawowe - RP). Warto zaznaczyć, że reguły RR są semantycznie równoważne regułom RP.

Dzięki połączeniu faktów właściwych i potrzebnych, rozszerzonych reguł, reguł mapujących i mechanizmu wnioskowania w przód możemy odpowiedzieć na zapytanie. Proces wnioskowania, pomimo że w przód, jest sterowany celem (zapytaniem).

Obie przedstawione metody regułowej realizacji zapytań zwiększają efektywność działania wnioskowania i udzielania odpowiedzi poprzez ukierunkowanie tych procesów na cel. Eksperymenty potwierdzające zwiększoną efektywność działania zostały zaprezentowane w Rozdziale 6.

4.3 Mapowanie Predykatów Ontologicznych na Dane Relacyjne

W celu umożliwienia integracji predykatów ontologicznych z danymi relacyjnymi opracowano metodę ich wzajemnego odwzorowania. Metoda bazuje na regułach mapujących następującej postaci:

$$\text{Zapytanie_SQL} \rightarrow \text{Predykat_bazowy}$$

Predykat bazowy to predykat, którego instancji nie można uzyskać w procesie wnioskowania, tylko należy ją wprowadzić wprost. Zazwyczaj będą to predykaty znajdujące się najniżej w hierarchii pojęć oraz relacji. Reguła mapująca, w wyniku uruchomienia, zadaje zapytanie SQL do relacyjnej bazy danych. Wynik zapytania jest dodawany do pamięci roboczej narzędzia wnioskującego w formie trójek RDF. Dopuszczalna forma zapytań została ograniczona do zapytań o następującej formie:

$$SELECT [R] FROM [T] < WHERE > < C, AND, OR >$$

gdzie:

- *R* to atrybuty (kolumny) - jeden lub dwa w zależności od arności predykatów,
- *T* to tabela, która zawiera kolumny *R*,
- *WHERE* opcjonalna klauzula do wprowadzenia ograniczeń,
- *C* ograniczenia w następującej formie < atrybut, operator, wartość >, na przykład: *Wiek > 18*,

- *AND* - komenda SQL umożliwiająca dodanie dodatkowych warunków (konjunkcja warunków).

W rozprawie zaproponowano oraz zaimplementowano metodę grupowania zapytań SQL celem zminimalizowania interakcji pomiędzy narzędziem wnioskującym a relacyjną bazą. Algorytm grupowania łączy zapytania dotyczące tego samego predykatu, ale różnych wartościowań zmiennych. Ten zabieg służy szybszej realizacji zapytań oraz kopiowania „w locie” danych do pamięci roboczej silnika wnioskującego.

Implementacja biblioteki SDL

Rozdział zawiera opis implementacji narzędzia SDL (ang. *Semantic Data Library*), w którym zostały zaimplementowane metody regułowej realizacji zapytań. Ponadto, narzędzie SDL ułatwia definiowanie odwzorowań pomiędzy predykatami ontologicznymi a danymi relacyjnymi. SDL umożliwia automatyczną generację wszystkich zbiorów reguł wymaganych w realizacji zapytań: reguł do wnioskowania w przód, reguł do wnioskowania wstecz, reguł rozszerzonych oraz reguł mapujących. Wszystkie generowane reguły są w postaci klauzul Horna. Językiem implementacji narzędzia SDL jest Java. Jako silnik wnioskujący wybrano narzędzie Jess, a interfejs bazodanowy obejmuje relacyjny serwer baz danych Microsoft SQL Server 2008.

SDL wykorzystuje narzędzie Pellet [Sirin 2007] do sprawdzania spójności ontologii oraz do wyliczenia hierarchii pojęć i relacji. Wykorzystywane jest również narzędzie OWL API [Horridge 2009] służące do obsługi plików OWL. SDL operuje na standardzie zapisu języka OWL 1.1 oraz reguł SWRL (przy czym dopuszczalne są tylko reguły w formie klauzul Horna).

Narzędzie SDL dostarcza również dwóch metod do transformacji ontologii w zbiór reguł. Pierwsza z metod (tzw. prosta) to przekształcenie wyliczonej hierarchii pojęć i relacji w reguły, oraz generacja reguł dla relacji odwrotnych i symetrycznych. Druga metoda jest zgodna z formalizmem Horn-*SHIQ* i generuje większy zbiór reguł, ale w pełni oddaje semantykę transformowanej ontologii (zakładając, że język ontologii nie wykracza siłą ekspresji poza ten formalizm). Obie metody transformacji szczegółowo zostały opisane w niniejszym rozdziale rozprawy.

Narzędzie SDL składa się z dwóch modułów: SDL-API (ang. *Application Programming Interface*) oraz SDL-GUI (ang. *Graphical User Interface*). Interfejs programistyczny SDL-API dostarcza wszystkich zaimplementowanych funkcji, włączając odczyt ontologii, generację reguł, metody zarządzania wnioskowaniem w narzędziu Jess oraz realizację zapytań. Natomiast aplikacja okienkowa SDL-GUI, korzystająca z SDL-API, umożliwia definiowanie mapowań, wyświetlanie wyliczonej hierarchii pojęć i relacji, wyświetlanie reguł, prezentowanie struktury relacyjnej bazy danych oraz generację zbiorów generowanych reguł w postaci skryptów zapisanych w języku Jess.

Narzędzie SDL może zostać pobrane (w celach niekomercyjnych) z następującej strony WWW <http://draco.kari.put.poznan.pl/>.

Eksperymenty obliczeniowe

W rozdziale zaprezentowano eksperymenty obliczeniowe z wykorzystaniem narzędzia SDL oraz bazy wiedzy MinOn. Testy wymagały skonstruowania relacyjnej bazy danych i jej wypełnienia wykorzystując zaimplementowany automatyczny generator danych. W kolejnym kroku zdefiniowano reguły mapujące pomiędzy bazą wiedzy MinOn a danymi relacyjnymi. Skonstruowano 5 zapytań testujących różne aspekty regułowej realizacji zapytań: wykorzystanie hierarchii pojęć i relacji, wykorzystanie stałych i/lub zmiennych, wykorzystanie reguł w szerszym zakresie (np. relacje symetryczne).

Wygenerowane bazy danych zawierają 10 tabel relacyjnych z informacją o: pracownikach, firmach, fakturach, podpisach na dokumentach, wyrządzeniu szkody bądź osiągnięciu zysku, dokumentach (nie) potwierdzających wykonanie pracy, produktach i usługach, obrocie pieniężnym, przepisach prawnych oraz o innych faktach, które pochodzą z zeznań w trakcie prowadzenia dochodzenia (np. że ktoś wiedział o czymś). W celu przeprowadzenia testów wygenerowano 3 relacyjne bazy danych różniące się liczbą dokumentów. Generacja danych i dokładne różnice pomiędzy bazami zostały szczegółowo przedstawione w niniejszym rozdziale rozprawy.

Testowe zapytania zostały wykonane z zastosowaniem następujących metod wnioskowania oraz transformacji:

- Metoda transformacji prostej:
 - metoda hybrydowa,
 - metoda rozszerzonych reguł,
 - metoda wnioskowania w przód w narzędziu Jess,
 - metoda wnioskowania wstecz w narzędziu Jess.
- Metoda transformacji Horn-*SHIQ* przy wykorzystaniu metody rozszerzonych reguł.

Wyniki czasowe wszystkich zapytań potwierdziły, że proponowane usprawnienie znacznie (w niektórych przypadkach kilkukrotnie) skraca czas potrzebny na uzyskanie odpowiedzi na zapytanie niż w przypadku zwykłej realizacji zapytań w narzędziu Jess. Ponadto wykazano, że składowanie danych na zewnątrz pamięci

robotycznej silnika wnioskującego przyczynia się do zwiększenia zbioru danych jaki może być „przepytany”. Dodatkowym atutem implementacji jest realizacja zapytań na bieżącym stanie relacyjnej bazy danych, co oznacza, że za każdym razem kiedy zapytanie jest wykonywane, SDL łączy się z bazą danych w celu udzielenia najbardziej aktualnej odpowiedzi. Ma to istotne znaczenie w przypadku, gdy baza danych ulega ciągłym zmianom.

ROZDZIAŁ 7

Wnioski

W rozprawie doktorskiej zaprezentowano oraz przetestowano dwie metody regułowej realizacji zapytań do bazy wiedzy o przestępstwach gospodarczych. Obie metody wykorzystują relacyjną bazę jako magazyn danych. Baza wiedzy, w postaci ontologii z regułami, opisuje dane przechowywane w relacyjnej bazie. W ten sposób osiągnięto główny cel pracy – zaprojektowano metodę regułowej realizacji zapytań z wykorzystaniem relacyjnej bazy danych oraz formalnie zdefiniowanej semantyki.

Najważniejsze rezultaty pracy są następujące:

- Opracowano nową modyfikację transformacji magic, która wprowadza tzw. rozszerzone reguły (RR). Metoda wykorzystuje mechanizmy wnioskowania w przód przy użyciu algorytmu Rete. Metoda pozwala na integrację relacyjnej bazy danych, narzędzia wnioskującego Jess oraz bazy wiedzy zapisanej w logice deskrypcyjnej Horn-*SHIQ* wraz z regułami w formie klauzul Horna. Metoda regułowej realizacji zapytań wykorzystująca rozszerzone reguły umożliwia kierowanie zapytań koniunkcyjnych do relacyjnej bazy danych w formie skierowanych grafów zbudowanych z predykatów ontologicznych. Metoda ta jest ogólna w tym sensie, że można ją zastosować we wszystkich narzędziach wnioskujących implementujących algorytm Rete. Podejście RR jest efektywniejsze niż hybrydowa metoda realizacji zapytań.
- Sformułowano bazę wiedzy o przestępstwach gospodarczych: sprzeniewierzeniu i praniu pieniędzy. Baza wiedzy została wyrażona w języku Horn-*SHIQ* wraz z regułami w postaci klauzul Horna. Baza wiedzy została zaimplementowana jako ontologia w językach OWL 1.1 oraz SWRL. Konstrukcja bazy wiedzy umożliwia odkrywanie działań przestępczych oraz określenie kwalifikacji prawnej popełnionych czynów.
- Zaproponowano metodę mapowania predykatów ontologicznych na dane relacyjne przy pomocy reguł zawierających zapytania SQL. W wyniku uzyskanych mapowań, użytkownik korzystających z metody RRZ, może formułować zapytania z koniunkcji predykatów ontologicznych. Jest to metoda dużo łatwiejsza niż w przypadku konstruowania zapytań w języku SQL.
- Opracowano metodę hybrydowego wnioskowania w procesie regułowej realizacji zapytań. Metoda steruje mechanizmami wnioskowania w przód oraz

wstecz zaimplementowanych w narzędziu Jess. Jest to metoda zależna od specyfiki działania tego silnika wnioskującego i z tego względu nie może być zastosowana w innych narzędziach.

- Zaimplementowano prototypowe narzędzie SDL, oferujące implementację opracowanych metod RRZ. Narzędzie składa się z dwóch części SDL-GUI oraz SDL-API. SDL-GUI dostarcza metod wspomagających mapowanie predykatów na dane relacyjne oraz generację różnych zbiorów reguł wykorzystywanych w obu metodach RRZ. SDL-API to moduł, który z powodzeniem może zostać wykorzystany w innych aplikacjach, gdzie wnioskowanie, realizacja zapytań oraz proste ich formułowanie są wymagane.
- Przeprowadzono eksperymenty obliczeniowe, które jednoznacznie potwierdzają, że proponowane usprawnienia w znaczący sposób zwiększają efektywność narzędzi działających na bazie algorytmu Rete. Ponadto udowadniają, że baza wiedzy o przestępstwach gospodarczych może ułatwić pracę śledczych i prokuratorów poprzez odkrywanie związków pomiędzy danymi za pomocą zadawanych zapytań. W ten sposób potencjalni użytkownicy metod zaproponowanych w rozprawie mogą przydzielać role osobom biorącym czynny udział w przestępstwie, a także otrzymać sugestie kwalifikacji prawnej czynów.

W trakcie prac nad rozprawą udało się zrealizować wszystkie przedstawione we wstępie cele. Zaprezentowane wyniki badań oraz eksperymentów stanowią punkt wyjścia dla dalszego rozwoju metod regułowej realizacji zapytań uwzględniających rozszerzone reguły. Ostatni rozdział pracy zawiera również inne kierunki badań obejmujących wizualizację reguł, czy też poddanie zaproponowanych metod testom w innych narzędziach wnioskujących oraz testom sprawności (ang. *benchmark*). Część tych prac została już rozpoczęta i opublikowana [Nowak 2012].

Bibliografia

- [Abiteboul 1995] Serge Abiteboul, Richard Hull and Victor Vianu. *Foundations of databases*. Addison-Wesley, 1995. (Cytowanie na stronie 15.)
- [Albrecht 2011] W.S. Albrecht, C.C. Albrecht, C.O. Albrecht and M.F. Zimbelman. *Fraud examination*. Cengage Learning, 2011. (Cytowanie na stronie 7.)
- [Baader 2003] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi and Peter F. Patel-Schneider, editors. *The description logic handbook: theory, implementation, and applications*. Cambridge University Press, New York, NY, USA, 2003. (Cytowanie na stronie 7.)
- [Bancilhon 1986] Francois Bancilhon, David Maier, Yehoshua Sagiv and Jeffrey D Ullman. *Magic sets and other strange ways to implement logic programs (extended abstract)*. In *Proceedings of the fifth ACM SIGACT-SIGMOD symposium on Principles of database systems, PODS '86*, pages 1–15, New York, NY, USA, 1986. ACM. (Cytowanie na stronie 7.)
- [Beckett 2004] Dave Beckett. *RDF/XML Syntax Specification (Revised)*. W3C recommendation, W3C, 2004. (Cytowanie na stronie 16.)
- [Beeri 1987] Catriel Beeri and Raghu Ramakrishnan. *On the Power of Magic*. In *Journal of Logic Programming*, pages 269–283, 1987. (Cytowanie na stronie 17.)
- [Bry 2007] François Bry, Norbert Eisinger, Thomas Eiter, Tim Furge, Georg Gottlob, Clemens Ley, Benedikt Linse, Reinhard Pichler and Fang Wei. *Foundations of Rule-Based Query Answering*. In Grigoris Antoniou, Uwe Aßmann, Cristina Baroglio, Stefan Decker, Nicola Henze, Paula-Lavinia Patranjan and Robert Tolksdorf, editors, *Reasoning Web*, volume 4636 of *Lecture Notes in Computer Science*, pages 1–153. Springer, 2007. (Cytowanie na stronie 2.)
- [Community 2012] JBoss Community. *Drools - The Business Logic integration Platform*. <http://www.jboss.org/drools/>, 2012. Accessed: 04/04/2013. (Cytowanie na stronie 3.)
- [Forgy 1982] C. L. Forgy. *Rete: A fast algorithm for the many pattern/many object pattern match problem*. *Artificial Intelligence*, vol. 19, pages 17–37, 1982. (Cytowanie na stronach 3 and 15.)

- [Gallaire 1978] Hervé Gallaire and Jack Minker, editors. Logic and data bases, symposium on logic and data bases, centre d'études et de recherches de toulouse, 1977, *Advances in Data Base Theory*. Plenum Press, 1978. (Cytowanie na stronie 7.)
- [Gómez-Pérez 2008] A. Gómez-Pérez, C. Suárez de Figueroa Baonza M. and B. Villazón. *NeOn Methodology for Building Ontology Networks: Ontology Specification, excerpt from NeOn Deliverable D5.4.1*. <http://www.neon-project.org>, 2008. (Cytowanie na stronie 10.)
- [Grosz 2003] Benjamin N. Grosz, Ian Horrocks, Raphael Volz and Stefan Decker. *Description logic programs: combining logic programs with description logic*. In Proceedings of the 12th international conference on World Wide Web, WWW '03, pages 48–57, New York, NY, USA, 2003. ACM. (Cytowanie na stronie 7.)
- [Hill 2003] Ernest Friedman Hill. *Jess in action: Java rule-based systems*. Manning Publications Co., Greenwich, CT, USA, 2003. (Cytowanie na stronach 3, 7 and 15.)
- [Horridge 2006] Matthew Horridge, Nick Drummond, John Goodwin, Alan Rector and Hai H Wang. *The Manchester OWL Syntax*. In In Proc. of the 2006 OWL Experiences and Directions Workshop (OWL-ED2006, 2006. (Cytowanie na stronie 16.)
- [Horridge 2009] Matthew Horridge and Sean Bechhofer. *The OWL API: A Java API for Working with OWL 2 Ontologies*. In OWLED, 2009. (Cytowanie na stronie 21.)
- [Horrocks 2004a] Ian Horrocks and Peter F. Patel-Schneider. *A proposal for an owl rules language*. In Proceedings of the 13th international conference on World Wide Web, WWW '04, pages 723–731, New York, NY, USA, 2004. ACM. (Cytowanie na stronie 7.)
- [Horrocks 2004b] Ian Horrocks, Peter F. Patel-schneider, Harold Boley, Said Tabet, Benjamin Grosz and Mike Dean. *SWRL: A semantic web rule language combining OWL and RuleML*. 2004. Accessed: 04/04/2013. (Cytowanie na stronie 12.)
- [Horrocks 2005] Ian Horrocks, Peter F. Patel-schneider, Sean Bechhofer and Dmitry Tsarkov. *OWL Rules: A Proposal and Prototype Implementation*. *Journal of Web Semantics*, vol. 3, pages 23–40, 2005. (Cytowanie na stronie 7.)

- [Hustadt 2005] Ullrich Hustadt, Boris Motik and Ulrike Sattler. *Data Complexity of Reasoning in Very Expressive Description Logics*. In IN PROC. IJCAI 2005, pages 466–471. Professional Book Center, 2005. (Cytowanie na stronie 7.)
- [Krötzsch 2008] Markus Krötzsch, Sebastian Rudolph and Pascal Hitzler. *Description Logic Rules*. In Malik Ghallab, Constantine D. Spyropoulos, Nikos Fakotakis and Nikos Avouris, editors, Proceedings of the 18th European Conference on Artificial Intelligence (ECAI'08), pages 80–84. IOS Press, 2008. (Cytowanie na stronie 7.)
- [Liang 2009] Senlin Liang, Paul Fodor, Hui Wan and Michael Kifer. *OpenRule-Bench: an analysis of the performance of rule engines*. In Proceedings of the 18th international conference on World wide web, WWW '09, pages 601–610, New York, NY, USA, 2009. ACM. (Cytowanie na stronie 2.)
- [Martinek 2008] Jacek Martinek. *Hydra case formal structural description*. Rapport technique, Poznan, Poland, 2008. (Cytowanie na stronach 9 and 10.)
- [Motik 2004] Boris Motik, Ulrike Sattler and Rudi Studer. *Query Answering for OWL-DL with Rules*. In Journal of Web Semantics, pages 549–563. Springer, 2004. (Cytowanie na stronach 7 and 15.)
- [Nowak 2012] Maciej Nowak, Jaroslaw Bak and Czeslaw Jedrzejek. *Graph-based Rule Editor*. In Hassan Aït-Kaci, Yuh-Jong Hu, Grzegorz J. Nalepa, Monica Palmirani and Dumitru Roman, editors, RuleML2012@ECAI Challenge and Doctoral Consortium at the 6th International Symposium on Rules, Montpellier, France, August 27th-29th, 2012, volume 874 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012. (Cytowanie na stronie 26.)
- [PricewaterhouseCoopers 2009] PricewaterhouseCoopers. *The Global Economic Crime Survey*. http://www.pwc.com/en_GX/gx/economic-crime-survey/pdf/global-economic-crime-survey-2009.pdf, 2009. Accessed: 04/04/2013. (Cytowanie na stronie 10.)
- [Sagiv 1984] Yehoshua Sagiv and Jeffrey D. Ullman. *Complexity of a top-down capture rule*. Rapport technique, Stanford, CA, USA, 1984. (Cytowanie na stronie 7.)
- [Salinger 2004] L.M. Salinger. Encyclopedia of white-collar & corporate crime. Numeéro t. 1 de Encyclopedia of White-collar & Corporate Crime. SAGE Publications, 2004. (Cytowanie na stronie 8.)

- [Sejm 1997] Polish Sejm. *Polish Penal Code*. <http://isap.sejm.gov.pl/Download?id=WDU19970880553&type=3>, 1997. Accessed: 04/04/2013. (Cytowanie na stronie 4.)
- [Sirin 2007] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur and Yarden Katz. *Pellet: A practical OWL-DL reasoner*. *Web Semant.*, vol. 5, no. 2, pages 51–53, June 2007. (Cytowanie na stronie 21.)

Alternatywna metoda dla MS Word to w ustawieniach autokorekty wstawić podwójne interpunkcje i wybrać zamienianie ich na pojedyncze. Jednak Wordowi nie ufam do końca, więc wolę to robić ręcznie po zakończeniu pracy. Ale i tak zachcę do spróbowania tej metody, bo może komuś będzie bardziej odpowiadać. 5. Dość często arkusze zapytań. Niektórzy klienci wymagają przesłania zapytań terminologicznych po zakończeniu całości tłumaczenia. Więc dodanie arkusza zapytań do odsyłanej paczki to mus. 6. Poprawki z własnej listy. Często w polowie tłumaczenia przychodzi oświecenie i fraza tłumaczona do tej pory tak i tak przestaje pasować i bardziej mi się podoba nowa wersja. Specjalnej uwagi wymaga opracowanie wiedzy socjokulturowej dla przedstawicieli krajów sąsiednich z tego względu, a także w krajach bliskich przestrzennie funkcjonują stereotypy zniekształcające (przesadnie upraszczające) wiedzę o Polsce, jej społeczeństwie i kulturze. Dlatego też uczenie języka polskiego w takich krajach ma sprzyjać lepszemu poznaniu kultury i realiów kraju sąsiedniego oraz przewycięć ukształtowane stereotypy.