

AKADEMIA GÓRNICZO-HUTNICZA

im. St. Staszica w Krakowie

WEAiE, Katedra Automatyki

Laboratorium Informatyki w Sterowaniu i Zarządzaniu

Przedmiot:
Inżynieria wiedzy

ZARZĄDZANIE WIEDZĄ (Knowledge Management)

wykonali:

Filip Mich (mini_fm@interia.pl)

Bartosz Kryza (kryza@student.uci.agh.edu.pl)

IV rok AiR

Wersja 1.0

Kraków, wrzesień 2003.

Spis treści

1	Wstęp	3
1.1	Potrzeba KM	3
1.2	Definicje KM	5
2	Analiza aktualnych rozwiązań KM	6
2.1	Ogólna charakterystyka	6
2.2	Przegląd systemów kompleksowych	8
2.2.1	IBM Lotus Notes	8
2.2.2	Pyton Enterprise	8
2.2.3	Hummingbird Enterprise	10
2.3	Przegląd rozwiązań częściowych	11
2.3.1	AskJeeves	11
2.3.2	VizServer (Inxight)	11
2.3.3	SmartDiscovery (Inxight)	11
2.3.4	Intelligent Miner for Text (IBM)	11
2.3.5	Hyperion Analyzer	12
3	Rozwiązania oparte o metody formalne	13
3.1	Wprowadzenie	13
3.2	Główne technologie	14
3.2.1	RDF - Resource Description Framework	14
3.2.2	DAML - DARPA Agent Markup Language	15
3.2.3	DQL - DAML Query Language	15
3.2.4	OKBC - Open Knowledge Base Connectivity	16
3.2.5	OWL - Web Ontology Language	16
3.2.6	KIF - Format Wymiany Wiedzy	16
3.2.7	Formal Concept Analysis	16
3.3	Narzędzia	18
3.3.1	Concept Explorer	18
3.3.2	ToscanaJ	18
3.3.3	KAON	18
3.3.4	On-To-Knowledge	20
3.3.5	Protege-2000	21
4	Podsumowanie	21
A	Zasoby w sieci	22

Streszczenie

Celem niniejszego opracowania jest przegląd metod i narzędzi służących zarządzaniu wiedzą w nowoczesnych korporacjach oraz nowatorskich technik formalnych rozwijanych przez środowiska akademickie. Poniższa praca jest luźną kompilacją pomysłów, idei i trendów związanych z aktualnymi trendami w dziedzinie KM (Knowledge Management), choć prace i artykuły wykorzystane przy tym opracowaniu często nie miały na pozór dużego związku z tym zagadnieniem.

1 Wstęp

1.1 Potrzeba KM

W dzisiejszych czasach globalnych korporacji i swobodnego dostępu do ogromnych ilości danych i informacji, wszystkie liczące się przedsiębiorstwa dostrzegają wyraźną potrzebę zapanowania nad nimi w sposób zorganizowany.

Niestety do niedawna większość przedsięwzięć mianowanych przez swych twórców jako rozwiązania KM, były niczym więcej niż "prostymi" pod względem teoretycznym systemami wymiany i katalogowania dokumentów. Dopiero niedawno największe korporacje takie jak Xerox Corporation, Hewlett-Packard, General Motors czy IBM dostrzegły potrzebę stworzenia bardziej kompleksowych rozwiązań, umożliwiających im zarządzanie nie tylko informacją, do czego sprowadzały się standardowe systemy wspomagające wymianę dokumentów, ale również wiedzą i to zarówno tą formalną, udokumentowaną w postaci opracowań pracowników czy zakupionych ze źródeł zewnętrznych ale również tak zwaną wiedzą potoczną, nigdzie nie dokumentowaną a jedynie wymienianą pomiędzy pracownikami często na nieformalnych spotkaniach po pracy.

Głównym problemem motywującym korporacje do poszukiwania i stosowania rozwiązań KM jest obserwacja, że pracownicy firmy nie mają optymalnego dostępu do potrzebnych im informacji i wiedzy innych pracowników. Często nie wiedzą oni nawet, że taka wiedza istnieje w obrębie tej samej organizacji. Doprowadza to do sytuacji, gdy jeden z zespołów czy pracowników powtarza pracę innego, często za przyzwoleniem swoich przełożonych, którzy również nie zdają sobie sprawy, że będzie to marnowanie czasu i pieniędzy.

Dodatkowym problemem jest fakt, że w każdej korporacji dochodzi do utraty wiedzy. Utrata ta jest spowodowana niedoskonałością ludzkiej pamięci, odchodzeniem pracowników czy po prostu zmian, które powodują, że wiedza raz zdobyta staje się nieaktualna.

Niestety, główny nacisk przy rozwijaniu systemów KM przez duże korporacje kładzony jest na rozwijanie procedur wymiany danych pomiędzy pracownikami czy wręcz archiwizowania spotkań i konferencji odbywających się poprzez media elektroniczne i umożliwianiu dostępu do nich, zamiast aktywnego wspierania i testowania badań w dziedzinie metod formalnych służących reprezentacji wiedzy, dostępu do niej oraz automatycznego wnioskowania wspomagającego pracowników.

Kolejnym powodem dla którego rośnie zapotrzebowanie na tego typu systemy jest konieczność nadążania za konkurencją. Podobnie jak w wyścigu Formuły 1, wskazanie

obrotomierza sprzed 20 minut jest mniej niż bezużyteczne, tak w przypadku nowoczesnych przedsiębiorstw o zasięgu globalnym aktualna i wiarygodna informacja i wiedza jest praktycznie bezcenna. Aby jednak możliwe było utrzymywanie tej wiedzy na bieżąco, konieczne jest również stałe uaktualnianie i rozwijanie metod zdobywania, zarządzania i dostępu do wiedzy i informacji.

Nie jest to jednak takie proste. Systemy KM oferowane przez *największych* rynku IT są bardzo różne i napewno bardzo dalekie od ideału. Podczas gdy dane rozwiązanie może przynieść ogromne korzyści jednej firmie drugą może doprowadzić do ruiny.

Jednakże wszyscy eksperci zgadzają się co do jednego. Głównym celem systemów KM stosowanych w dużych przedsiębiorstwach powinno być nie tylko umożliwienie pracownikom dzielenia się wiedzą, ale wręcz zmotywowanie ich do takiej aktywności. Tutaj jednak problemem nie jest już technologia zaprzęgnięta do tego celu, ale atmosfera i psychologia pracy. Pracownicy muszą czuć, że powinni dzielić się wiedzą ze swoimi współpracownikami. Tutaj można przedstawić jako przykład międzynarodową firmę konsultingową Ernst&Young, która wprowadziła na wielką skalę w swoich filiach system IBM Lotus Notes, co jednak nie przełożyło się w żaden sposób na poprawę wydajności pracy, gdyż pracownicy bali się upubliczniać swoją wiedzę, w obawie o przywłaszczenie sobie przez kogoś innego danej informacji i uzyskanie za nią pochwały.

Naszym zdaniem, między innymi tym należy tłumaczyć fakt, że specjaliści od KM w korporacjach, zazwyczaj ludzie bez specjalistycznego wykształcenia informatycznego, boją się inwestować w badania nad nowymi technologiami, a raczej stawiają na zmiany organizacji pracy i nowe sposoby motywowania pracowników do dzielenia się własnymi doświadczeniami.

Davenport i Prusak w swojej książce pt. "Working Knowledge", która stała się *biblią* dla wszystkich osób związanych z tematyką zarządzania wiedzą, twierdzą, że w przypadku wdrażania systemów zarządzania wiedzą 80 proc. nakładów ponosi się na tworzenie kultury organizacyjnej, a jedynie 20 proc. na systemy informatyczne

Jednak według nas nie można się opierać jedynie na rozwiązaniach z dziedziny zarządzania. Należy je podeprzeć na rozwiązaniach informatycznych, formalnych lub pół-formalnych, które w połączeniu ze zmianą filozofii pracy w danej organizacji mogą doprowadzić do drastycznej poprawy wydajności. Dobrym negatywnym przykładem może tutaj być przypadek NASA, a dokładnie Kennedy Space Center z Florydy. Analiza stanu instytucji wykonana na początku lat 90-tych pokazała, że większość badań, nawet tych podstawowych, wykorzystywanych przy lotach kosmicznych, odbywało się przez zatrudnianie innych instytutów badawczych na kontrakt, ponieważ ogromna większość wybitnych specjalistów, którzy pracowali tam w czasie zimnej wojny odeszło na emerytury, nie pozostawiając po sobie wiedzy, którą posiadali. Wówczas to kierownictwo NASA podjęło decyzję o reorganizacji i stworzenia systemu zarządzania wiedzą, który miałby odbudować kadrę i nie dopuścić do takiej sytuacji w przyszłości. Rezultatem tego przedsięwzięcia było zbudowanie bardzo prymitywnego systemu wymiany informacji, który właściwie sprowadzał się do zbioru specjalizowanych grup dyskusyjnych, na które pracownicy mieli wysyłać swoje odkrycia jak i problemy. Dziś takie rozwiązanie wydaje się dość naiwne, a krytycy NASA po katastrofie Columbi, podkreślali, że główną przyczyną katastrofy nie była wada konstrukcyjna a problemy z organizacją pracy badawczej i zarządzaniem projektami w

NASA.

1.2 Definicje KM

Pomimo, że badania nad obecnymi formami KM trwają już co najmniej kilkanaście lat, nie istnieje żadna oficjalna i wspólna dla ludzi zajmujących się tą dziedziną definicja. Mówi się wręcz, że na 20 specjalistów od KM istnieje 30 definicji.

Jednak, aby dokonać próby przybliżenia czym powinno być KM, użyjemy najpierw trzech podstawowych definicji:

- **dane** - jest to zbiór obserwacji, faktów nie posiadających interpretacji, a jedynie pewną reprezentację, np. dokument tekstowy, e-mail, film etc...
- **informacja** - są to zinterpretowane dane, które wpływają na podejmowane decyzje. Tak więc aby jakieś dane stały się informacją trzeba wykonać pewną pracę intelektualną nad selekcją i asymilacją tych danych do określonych sposobów myślenia w danej organizacji.
- **wiedza** - jest to zbiór informacji, które zmieniają strukturę i procesy przedsiębiorstwa
- **mądrość** - powstaje kiedy zrozumie się ogólne wzorce tworzące wiedzę, wzorce które są uniwersalne, niezależne od kontekstu

Można również zdefiniować trzy procesy zachodzące pomiędzy tymi trzema poziomami informacji:

- **obróbka danych** - odnajdywania danych w różnych zasobach zarówno wewnętrznych jak i zewnętrznych, walidacja ich poprawności i przechowywanie ich w odpowiedni sposób aby były szybko i łatwo dostępne
- **zarządzanie informacją** - transformacja danych do reprezentacji ułatwiających ich analizę nie tylko przez ludzi ale również przez systemy komputerowe
- **budowanie wiedzy** - odnajdywanie w grupach informacji związków pozwalających na tworzenie wzorców zachowań w zależności od sytuacji
- **nabywanie mądrości** - odnajdywanie uniwersalnych wzorców w wiedzy którą się posiada i do której ma się dostęp

Na przykład, stwierdzenie *"Jeżeli dana fabryka naszego przedsiębiorstwa umieszczona w USA nie przynosi oczekiwanych zysków, należy ją zamknąć a otworzyć nową w innym państwie, w którym jest tańsza siła robocza i mniejsze podatki."* można zaliczyć do poziomu wiedzy, podczas gdy na podstawie posiadanych informacji zostanie podjęta decyzja gdzie dokładnie przenieść daną fabrykę. Mądrości nie można jednak wskazać na tak małym przykładzie, gdyż powstanie ona dopiero po przeanalizowaniu wielu podobnych sytuacji, ich korzyści, ryzyka i wynikłych konsekwencji.

⁰Zakładamy, że wielkie amerykańskie korporacje nie liczą się z tym, że zamykając fabrykę w małym miasteczku zabierają automatycznie miejsca pracy większości jego mieszkańców...

Tak więc można powiedzieć, że informacja odpowiada nam na pytania *co?*, *gdzie?*, *kiedy?*, wiedza odpowiada na pytanie *jak?*, natomiast mądrość odpowiada na pytanie *dłaczego?*.

Tutaj pojawia się jednak następujący problem. Żaden z tych elementów, czyli dane, informacja, wiedza i mądrość nie znaczy wiele bez pozostałych. Wiedzę należy ciągle uaktualniać o nowe informacje a mądrość o nową wiedzę. Tak więc elementy tworzą cykl, jako że zawsze ma się tylko niepełną ilość danych, niekompletny zbiór informacji, niewystarczające zrozumienie zależności pomiędzy informacjami tworzącymi naszą wiedzę oraz małe pojęcie o uniwersalności posiadanej przez nas wiedzy. Tylko więc poprzez stałe uaktualnianie i weryfikowanie każdego z tych elementów równocześnie możemy się rozwijać.

Na podstawie tych definicji widać, że dobry system KM powinien ułatwiać czy wręcz automatyzować każdy z tych poziomów, może poza poziomem czwartym, który wydaje się dla reprezentacji formalnej jeszcze trochę zbyt skomplikowany.

Obecnie działające rozwiązania KM są to jednak rozwiązania automatyzujące jedynie dwa pierwsze poziomy czyli zdobywanie i filtrowanie danych oraz strukturyzację ich w informacje. Budowanie wiedzy natomiast jest pozostawione analitykom. Powoduje to sytuację, w której jedynie informacje są zachowywane trwale, a wiedza analityków z różnych względów jest w tracona w czasie.

Można powiedzieć, że obecnie systemy wyszukiwania i ekstrakcji danych oraz strukturyzowania ich w informacje są dość zaawansowane i mają duże praktyczne możliwości.

Naszym zdaniem więc badania nad nowoczesnymi systemami zarządzania wiedzą powinny przede wszystkim koncentrować się na sposobach reprezentacji i przechowywania wiedzy w sposób jak najbardziej przypominający ludzką pamięć. Takimi rozwiązaniami były od dawna systemy ekspertowe. Jednak obecnie są one wypierane w środowiskach naukowych przez bardziej ogólne i uniwersalne systemy oparte na ontologiach formalnych. Systemy ontologiczne pomimo, że wciąż są dość słabo ustandaryzowane i trudne w implementacji umożliwiają wymianę wiedzy pomiędzy dziedzinami, podczas gdy klasyczne systemy ekspertowe były praktycznie ścisłym obrazem wiedzy grupy ekspertów.

Nadzieja istnieje w przedsięwzięciach, które mają na celu standaryzację i formalizację wiedzy, głównie opartej na ontologiach np. w postaci koncepcji takie jak FCA (*ang.* Formal Concept Analysis) czy badania w ramach projektu EU-IST Semantic Web.

2 Analiza aktualnych rozwiązań KM

2.1 Ogólna charakterystyka

W przypadku systemów KM stosowanych w przedsiębiorstwach stosowane są one przede wszystkim do celów zarządzania bardziej niż do przechowywania wiedzy np. o technologii produkcji. Wynika to prawdopodobnie z tego, że wiedza technologiczna jest od dawna w jakiś sposób standaryzowana i dokumentowana. Tak więc rozwiązania rozumiane jako KM są stosowane głównie do zarządzania kompetencjami, zarządza-

nia ludźmi, zarządzaniem dokumentami i przepływem informacji czy tworzenia bazy wiedzy o klientach lub utrzymywania relacji z klientami (CRM - Customer Relationship Management).

Można wyróżnić następujące komponenty składające się na kompleksowe rozwiązanie KM w dużym przedsiębiorstwie:

1. **Portal korporacyjny** - zapewnia pojedynczy punkt dostępu do wszelkich danych i aplikacji oraz do grup użytkowników w celu wymiany informacji czy doświadczeń
2. **Silnik wyszukiwawczy** - pozwala na przeszukiwanie zarówno ustrukturyzowanych (w przypadku obecnych systemów są to raczej "zwykłe" bazy danych) jak i nieustrukturyzowanych informacji (dokumentów, e-maili, grup dyskusyjnych, plików wideo itp...)
3. **Klasyfikator** - dokonuje klasyfikacji i indeksacji informacji i wszelkich dokumentów pojawiających się w systemie umożliwiając późniejsze ich odnajdywanie, powinien być skonfigurowany pod potrzeby danego przedsiębiorstwa
4. **Archiwum elektroniczne** - powinno umożliwiać przechowywanie wszelkiego rodzaju dokumentów oraza łatwy i szybki dostęp do nich
5. **Środowisko współpracy** - powinno umożliwiać tworzenie wirtualnych grup pracowników pracujących nad danym projektem, zapewniać im możliwość szybkiej wymiany danych (czaty, wideo-konferencje, bezpieczny i współdzielony dostęp do wspólnych dokumentów)
6. **Wspólnoty pracowników** - tworzenie grup pracowników o wspólnych zainteresowaniach, umożliwiające im wymianę doświadczeń i nowych pomysłów (np. poprzez wewnętrzne grupy dyskusyjne) i archiwizowanie ich aktywności
7. **Systemy agentów** - powinny zapewniać ciągłe przeszukiwanie zarówno korporacyjnego intranetu jak i sieci globalnej w poszukiwaniu interesujących danego pracownika informacji i automatycznie go o tym powiadamiać
8. **Wizualizacja** - moduły z tej kategorii powinny umożliwiać użytkownikom łatwe, wręcz intuicyjne poruszanie się po zgromadzonych w repozytoriach danych za pomocą pewnych struktur zasobów (konceptji) powiązanych ze sobą łatwymi do zrozumienia relacjami (np. autor, dziedzina itp...)
9. **System wnioskujący** - ten, potencjalnie najciekawszy pod względem teoretycznym, komponent powinien umożliwiać użytkownikom odpowiadaniu na pytania użytkowników (najlepiej w języku naturalnym) np. *Co to jest komputer?* oraz potwierdzanie hipotez przez nich stawianych np. *Laptop jest komputerem*. Niestety obecnie brak jest standardowego formatu reprezentacji danych wraz z informacją o ich znaczeniu co umożliwiałoby efektywne wnioskowanie np. przy pomocy logiki I-go rzędu. Nadzieją jest tutaj upowszechnienie się takich formatów jak RDF czy DAML+OIL.

Głównym naszym zdaniem problemem jest tutaj to, że firmy reklamujące się jako oferujące kompleksowe rozwiązania KM tworzą przykrojone na miarę danego przed-

siębiorstwa systemy wymiany dokumentów i informacji między pracownikami najczęściej opartymi o pewnego rodzaju portal intranetowy. Takie rozwiązania cechują się tym, że nie istnieje standard w którym przechowywana jest ta wiedza co praktycznie uniemożliwia wymianę tych danych pomiędzy różnymi rozwiązaniami.

Kolejnym problemem jaki tu się pojawia jest to, że wiedza jest strukturyzowana najczęściej automatycznie na podstawie pewnych słów kluczowych pojawiających się w dokumentach a nie jest od podstaw strukturyzowana co naszym zdaniem znacznie poprawiłoby skuteczność wyszukiwania i wnioskowania na podstawie zgromadzonej bazy wiedzy.

2.2 Przegląd systemów kompleksowych

2.2.1 IBM Lotus Notes

Produkt firmy Lotus/IBM jest znany na rynku od dość dawna jako efektywne narzędzie pracy grupowej. Od dłuższego czasu środowisko Notes było rozwijane pod względem możliwości zarządzania wiedzą. Środowisko to nie jest jednak gotowym produktem a bardziej platformą do tworzenia specyficznych rozwiązań. Wadą tego rozwiązania jest koszt przystosowania tego środowiska do konkretnego przedsiębiorstwa, jednak zaletą jest umożliwienie łatwej integracji pomiędzy różnymi systemami opartymi o tą platformę.

Główne składniki kwalifikujące Notes do miana systemu KM to:

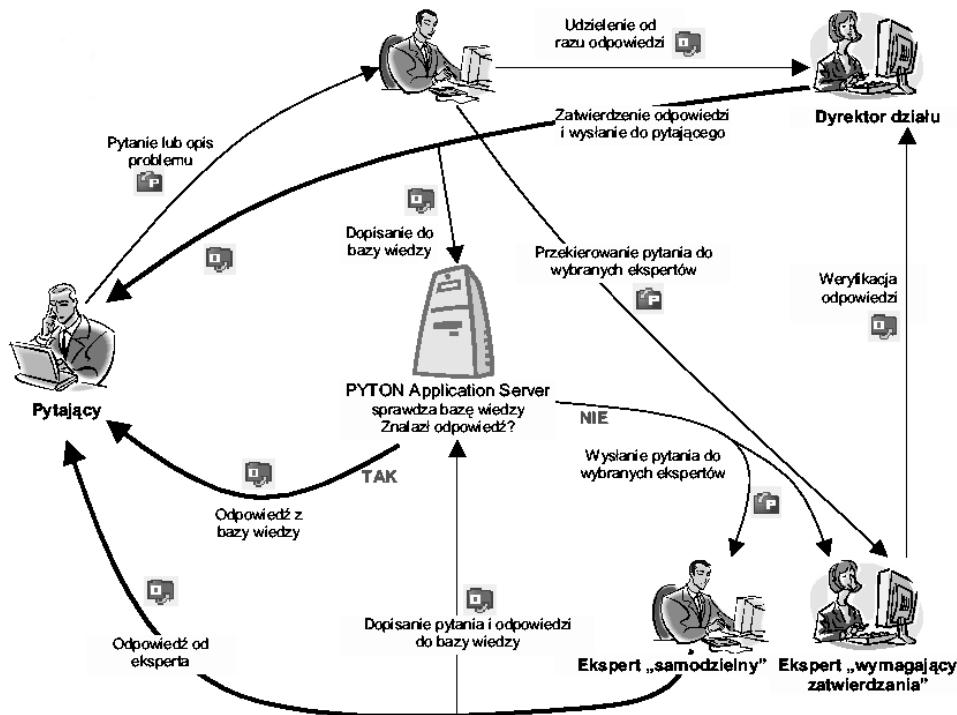
- **Baza dokumentów** - pozwala na przechowywanie wszelkiego rodzaju dokumentów oraz meta-informacji o tych dokumentach zarówno służących przy przeszukiwaniach typu *full-text search* oraz zdefiniowanych przez autora dokumentu słowach kluczowych.
- **Bezpieczeństwo** - w miarę uniwersalny system zabezpieczania informacji w systemie pozwala na kontrolę dostępu do danych w systemie na podstawie ich uprawnień.
- **Mapy treści** - przydzielanie dokumentom i informacjom kontekstu co ma ułatwiać wyszukiwanie i analizowanie danych przez pracowników
- **Metryki** - przechowywanie historii dokumentów, kto z nich korzystał, jak je oceniał, ewentualne poprawki i uwagi pracowników

Przykłady dużych udanych rozwiązań opartych o Lotus Notes to np. system do rozproszonej współpracy zespołów pracowników w British Petroleum czy stworzenie ogromnej bazy wiedzy dla Andersen Consulting. Niestety również wiele przedsiębiorstw twierdzi, że zastosowanie Lotus Notes nie jest wystarczające aby znacząco zwiększyć wykorzystanie kapitału jakim jest wiedza w nowoczesnym przedsiębiorstwie.

2.2.2 Pyton Enterprise

Pyton jest systemem KM tworzonym przez polską firmę Pyton Software. Jak piszą jego autorzy system pozwala na katalogowanie, archiwizowanie, udostępnianie oraz

transfer "Wiedzy Firmowej". Architektura systemu Pyton oparta jest o model pytanie-odpowiedź umożliwiający pracownikom zadawanie pytań do systemu w języku naturalnym. System próbuje znaleźć w systemie gotową odpowiedź na podobne pytanie i zwraca ją pytającemu. Jeżeli nie uda mu się znaleźć zadowalającej odpowiedzi system przekierowuje to pytanie do zespołu ekspertów, którzy odpowiadając na pytanie automatycznie uaktualniają bazę wiedzy systemu.



Rysunek 1: Architektura systemu Pyton

System Pyton składa się z następujących modułów:

- **OneQuestion** - posiada parser języka naturalnego do analizy pytań użytkowników, wykorzystujący słownik około 1,5 mln słów oraz analizuje statystyczny rozkład fraz w tekście, stosuje wnioskowanie rozmyte do odsiewania "błędów" w pytaniach, analizuje historię aktywności użytkowników, dokonuje klasyfikacji treści oraz umożliwia wizualizację wyników.
- **Collaborative Services** - umożliwia zarządzanie zasobami ekspertów z różnych dziedzin wiedzy wykorzystywanej w przedsiębiorstwie, automatyczny dostęp do ich wiedzy oraz ocenianie odpowiedzi i tworzenie w ten sposób klasyfikacji ekspertów. Umożliwia również generowanie raportów pokazujących w jakim stopniu wykorzystywany jest system przez poszczególnych pracowników.
- **K-Finder** - odpowiada za indeksowanie odpowiedzi i wyszukiwanie. Indeksowanie w systemie Pyton polega na opisanie zasobu za pomocą zbioru fraz klucz-

zowych a następnie obliczenie miar statystycznych odzwierciedlających korelację z zasobami wewnętrznymi takimi jak profile ekspertów, skatalogowane odpowiedzi itp...

- **Logic** - jest to regułowy system ekspertowy, wybierający najlepsze możliwe rozwiązania zadanych problemów takich jak dobór produktów dla klienta czy analiza wymagań klienta. Wykorzystuje algorytm propagacji wstecznej co oznacza, że system stawia hipotezy, które następnie udowadnia. W przypadku powodzenia dodaje je do bazy wiedzy, a w przypadku niepowodzenia odrzuca.
- **Q-Maker** - pozwala na integrację systemu Pyton z programem pocztowym w celu lepszej segregacji e-maili otrzymywanych z systemu. Również zapytania do systemu można kierować poprzez odpowiednio zredagowane e-maile.

System Pyton najlepiej nadaje się do zastosowań związanych z budowaniem bazy CRM, wprowadzaniu nowych produktów na rynek i zwiększaniem efektywności pracowników w wyszukiwaniu informacji na podstawie opisu problemu z jakim mają do czynienia co ma tą zaletę, że użytkownik nie musi dokładnie wiedzieć jakie informacje są mu potrzebne a system powinien mu je dostarczyć w oparciu o podobne przypadki wcześniej rozwiązane przez pracowników firmy.

Do wad tego rozwiązania można zaliczyć małą uniwersalność oraz brak w miarę ścisłej reprezentacji semantycznej przechowywanych dokumentów.

2.2.3 Hummingbird Enterprise

Ten system stanowi przykład jednego z najbardziej kompleksowych rozwiązań typu KM będącym jednocześnie rozwiązaniem dość uniwersalnym. System Hummingbird Enterprise składa się z następujących komponentów:

- **Portal** - internetowy serwis zapewniający dostęp do wszystkich pozostałych komponentów
- **Document Management** - jest to zintegrowany system przechowywania wszelkich rodzajów dokumentów i plików binarnych, pozwalający na konfigurację sposobu indeksowania tych danych, praktycznie niezależny od platformy i użytego systemu baz danych
- **Records Management** - przechowuje historię obiegu dokumentów, czyli kto, kiedy, jak korzystał z jakich dokumentów
- **Knowledge Management** - zawiera między innymi uniwersalny silnik wyszukiwawczy pozwalający na przeszukiwanie zasobów wewnętrznych, sieci WWW a także innych podobnych systemów w tym m.in. Lotus Notes. Dodatkowo posiada konfigurowalne systemy agentowe umożliwiające powiadamianie pracowników o nowych informacjach, które mogą być im potrzebne
- **Collaboration** - umożliwia tworzenie rozproszonych grup projektowych, organizowanie dla nich wspólnych "przestrzeni roboczych oraz komunikację. Pozwala również na włączanie do pracy nad projektami osób zewnętrznych: klientów, dostawców czy ekspertów.

- **Business Intelligence** - ułatwia analizę danych zgromadzonych w systemie oraz w zewnętrznych hurtowniach danych, systemach ERP oraz systemów OLAP, poprzez wizualizację na wielu poziomach szczegółowości
- **Extract-Transform-Load** - pozwala na projektowanie i organizację przepływu danych przy pomocy narzędzi wizualnych minimalizując porzebę programowania przy integracji systemu Hummingbird z systemami zewnętrznymi

2.3 Przegląd rozwiązań częściowych

Ten podrozdział ma na celu przedstawienie produktów wyspecjalizowanych w wykonywaniu jednego z zadań kompleksowego systemu KM. Ze względu na liczbę tych pozycji, są one omówione pokrótce.

2.3.1 AskJeeves

AskJeeves produkowany przez AskJeeves Inc. jest jednym z najbardziej zaawansowanych obecnie na rynku silników wyszukiwawczych opartych o język naturalny. Posiada również automatyczne sprawdzanie składni wpisanych wyrazów oraz posiada bardzo efektywny system klasyfikacji wyszukanych stron.

2.3.2 VizServer (Inxight)

VizServer oferuje zaawansowane techniki wizualizacji dużych ilości danych. Jego główne metody wizualizacji to:

- **StarTree** - oferuje interfejs graficzny który wg. autorów przyspiesza efektywne wyszukiwanie informacji o ponad 60 procent. Dane są wyświetlane jako drzewa gwiazdy połączonych koncepcji co ułatwia żagnieżdżanie się w interesujących danych
- **TableLens** - ułatwia odnajdywanie trendów, korelacji czy wzorców w danych, informacjach klientów czy konkurencji.

2.3.3 SmartDiscovery (Inxight)

Jest to produkt oferujący automatyczne wydobywanie danych z dokumentów tekstowych i ich organizację. Z interesujących właściwości można tutaj wyróżnić wsparcie dla 26 języków. System może być łatwo zintegrowany z większością głównym systemów zarządzających informacją, takimi jak Notes, Documentum, itp...

2.3.4 Intelligent Miner for Text (IBM)

System ten jest prawdopodobnie najbardziej rozbudowanym systemem służącym do wydobywania danych z dokumentów tekstowych. W tym celu wykorzystuje zaawansowane techniki statystyczne. Wykorzystuje i wspiera również oparty o XML standard PMML (*ang. Predictive Modeling Markup Language*) opracowany przez Data Mining

3 Rozwiązania oparte o metody formalne

3.1 Wprowadzenie

Formalna reprezentacja wiedzy wydaje się rzeczą bardzo trudną do osiągnięcia. Choć coraz więcej wiemy o tym jak działa mózg, jak przechowywane są w nim informacje nie bardzo wiadomo jak wykorzystać to do przechowywania wiedzy w komputerach.

Co jednak można zauważyć w badaniach nad ludzkim mózgiem, pamięcią, procesami poznawczymi i powstawaniem języka naturalnego jest zgoda naukowców z różnych dziedzin co do tego jaki charakter mają te wszystkie procesy zachodzące w mózgu. Miłośnicy według najnowszych naukowych trendów można wywnioskować, że główną operacją zachodzącą w mózgu, odpowiedzialną za prawie wszystkie jego procesy jest *kategoryzacja*. Kategoryzacja jest podstawą zarówno wnioskowania, zapamiętywania oraz powstania języków naturalnych [Lee96]. Nowe kategorie tworzone są przez cały czas poprzez rozszerzanie czy uaktualnianie starych. Natomiast konkretnym kategoriom są przypisywane przez mózg pewne konstrukcje językowe. Takie podejście jest sprzeczne z dawnym poglądem, że myślenie jest procesem związanym z przetwarzaniem języka naturalnego. Już nawet Einstein powiedział w [Ein54]: "Słowa czy język, zapisane czy wypowiedziane, nie wydają się mieć jakiegokolwiek związku z mechanizmem myślenia... Czy nie zdażyło się każdemu z nas próbować sobie bezskutecznie przypomnieć słów do wyrażenia czegoś co było dla nas w pełni oczywiste".

Tutaj można więc wysnuć wniosek, że ludzkie myślenie jest wręcz ograniczane przez język jakiego używamy do komunikacji. Powoduje to czasami, że człowiek dopóki sam czegoś nie zrozumie to nawet wielokrotne tłumaczenie mu tego niewiele pomoże.

Poniższa lista procesów zachodzących w mózgu pochodzi z książki [Lak87]:

- Myśl nie jest wynikiem ale procesem. Dokładniej jest to interakcja wielu procesów z których centralnym jest proces kategoryzacji.
- Wnioskowanie jest oparte na funkcjach kategoryzacji, które tworzą relacje między obiektami.
- Ludzkie myślenie nie może być ograniczone do zamkniętych, ściśle określonych z góry kategorii. Kategorie muszą przez cały czas ewoluować dzięki przyswajaniu nowych informacji i poprzez doświadczenie.
- Nie można również stworzyć zamkniętego zbioru funkcji kategoryzujących. Muszą one powstawać i adaptować się do nowych bodźców z otoczenia.
- Ludzkie rozumowanie w większości przypadków nie jest oparte na logice. Nie chodzi tu jednak o to że jest nielogiczne, po prostu jest ekstra-logiczne (...)
- Sam proces kategoryzacji jest procesem nabytym przy narodzinach i ewentualnie ulega pewnej ewolucji poprzez nowe doświadczenia i potrzebę przeswojenia nowej wiedzy.

Można więc powiedzieć, że wiedza w mózgu jest reprezentowana jako powiązania pomiędzy obiektami. Nie można jednak traktować tych powiązań całkiem ściśle jako matematycznych relacji funkcyjnych pomiędzy obiektami, między innymi dlatego, że "wynikzastosowania takiej funkcji do danego obiektu ściśle zależy od samego obiektu,

od tego do jakiej należy kategorii. Można więc powiedzieć, że jednostką wiedzy jest przydzielenie obiektu do jakiejś kategorii. Zbiór obiektów należących do tej samej kategorii, choć oczywiście mogą one jednocześnie należeć do innych kategorii, nazywamy konceptem. Tak więc koncept (podstawowe pojęcie Formalnej Analizy Konceptualnej) jest zbiorem obiektów dzielących wspólne cechy, na podstawie których nasz mózg łączy je we wspólne zbiory. Zbiór obiektów dzielących np. pewne 2 cechy, jest ekstensją tego konceptu a te 2 cechy stanowią intensję tego konceptu.

Tak więc naszym zdaniem, zamiast koncentrować się na symulowaniu działania mózgu w sensie biologicznym (sieci neuronowe) co ze względu na jego złożoność będzie jeszcze długo niemożliwe, lepiej skoncentrować się naszym zdaniem na symulowaniu sposobu w jaki przyswajane i przechowywane są informacje w mózgu. Tutaj pojawia się jednak problem, że ludzie nie potrafią wymieniać między sobą informacji w takiej postaci. Muszą najpierw zamienić (nie koniecznie świadomie) w możliwie dobrą reprezentację za pomocą języka naturalnego, a następnie druga osoba musi na podstawie tej reprezentacji słownej stworzyć własną reprezentację konceptualną. Ten proces jest analogiczny z wymianą plików graficznych przy użyciu jakiejś metody stratnej kompresji. Jeżeli ktoś będzie miał dużo czasu i umiejętności pedagogicznych na wyjaśnienie drugiej osobie pewnej części własnej wiedzy to ta druga osoba prawdopodobnie stworzy sobie strukturę konceptualną wystarczająco podobną do tej którą stworzyła sobie pierwsza osoba. Jeżeli to będzie tylko chwila to różnice mogą uniemożliwić dalszą komunikację (np. zaliczenie kolokwium). Problem jest otwarty, ale raczej nie należy liczyć na to żeby ludzie zaczęli porozumiewać się inaczej niż za pomocą języka naturalnego. Pojawia się jednak szansa, że tworząc komputerowe systemy wiedzy oparte o metody konceptualne, nawet uproszczone, to dzięki temu że komputery mogą bez przeszkód wymieniać całe struktury konceptualne w oryginalnej postaci, będzie możliwe dużo bardziej efektywne przechowywanie wiedzy i jej wykorzystywanie za pomocą komputerów niż miało to miejsce dotychczas.

Największe znane nam badania z tej dziedziny są tworzone jako część projektu SemanticWeb. Tworzone są tam systemy pozwalające na przechowywanie, wymianę i wnioskowanie oparte na podstawie ontologii.

3.2 Główne technologie

3.2.1 RDF - Resource Description Framework

RDF jest standardem opracowywanym przez W3C. Głównym celem tego standardu jest umożliwienie zawarcia w dokumentach internetowych informacji semantycznych "zrozumiałych" przez programy. RDF jest oparty na standardzie XML, co pozwala na definiowanie różnych zbiorów metadanych w zależności od konkretnej dziedziny w jakiej jest stosowany. Można to osiągnąć poprzez wykorzystywanie XML przestrzeni nazw (XSD - XML Schema Definiton). Jedną z takich standardowych przestrzeni nazw jest Dublin Core, znany od dawna zbiór metadanych dla stron WWW.

Podstawowym elementem składni RDF jest trójka: podmiot - *rdf:Subject*, obiekt - *rdf:Object* i orzeczenie - *rdf:Predicate*. Można to określić, że podmiot jest połączony z obiektem poprzez relację własności. Według zaleceń W3C każdy z tych elementów

powinien być reprezentowany przez kwalifikowany URI, czyli URI z dodanym opcjonalnie tekstem po znaku #.

Główną zaletą takiego rozwiązania jest jego prostota. Dzięki ściśle określonej, prostej strukturze predykatów opisujących dany obiekt możliwe jest łączenie wielu fragmentów w większe części i korzystanie z nich przy użyciu standardowych algorytmów grafowych.

Ze względu na zbyt ubogą możliwość reprezentacji struktur danych za pomocą samego RDF W3C rozszerzyło RDF o kilka nowych konceptów takich jak *rdf:subPropertyOf*, *rdfs:Container* umożliwiające definiowanie dodatkowo relacji pomiędzy strukturami danych.

3.2.2 DAML - DARPA Agent Markup Language

Ze względu na zbyt daleko idącą prostotę standardów RDF i RDFS widoczna stała się konieczność rozszerzenia możliwości definiowania klas i właściwości. Tak powstał standard DAML, rozwijany przez amerykańską rządową agencję DARPA. Dzięki szybkiemu połączeniu DAML ze standardem OIL (Ontology Inference Layer) powstał bardzo silny i uniwersalny standard DAML+OIL.

DAML+OIL wykorzystuje RDF i RDFS do definiowania prostych klas i własności, natomiast dodaje do tego między innymi większą kontrolę nad typami danych, definiowanie szerszych związków między własnościami np. możemy zdefiniować dwie nową własność jako identyczną z już istniejącą przy wykorzystaniu *daml:samePropertyAs*. Możemy również definiować własności jako unikalne - *daml:uniqueProperty*, odwrotne - *daml:inverseOf* czy przechodnie - *daml:TransitiveProperty*. Poza tym pozwala na definiowanie list (*daml:collection*) czy wycień (*daml:enumeration*).

Dzięki takim cechom DAML+OIL umożliwia coś co powinno umożliwić automatyczne wnioskowanie na podstawie zdefiniowanych ontologii, dzięki czemu takie czynności jak wyszukiwanie dokumentów czy dowodzenie hipotez w danej dziedzinie powinny wznieść systemy KM na wyższy jakościowo poziom.

3.2.3 DQL - DAML Query Language

DQL jest językiem/protokołem umożliwiającym agentom programowym odpytywanie bazy wiedzy przechowywanej w standardzie DAML+OIL. Zapytanie języka DQL zawiera zdania wyrażone w języku DAML+OIL, w których pewne stałe zostały zastąpione przez zmienne. Odpowiedź zawiera po prostu możliwe kombinacje stałych spełniające dane zapytanie gdy podstawią się je pod zmienne. Jest to więc podobne rozwiązanie jak w każdym języku logiki I-go rzędu np. Prologu. Tutaj dodano jednak możliwość określania "ważności zmiennych". Tak więc zmienne mogą być *must_bind*, *may_bind* oraz *dont_bind*.

Podobnie jak DAML+OIL - DQL jest również oparty na XML'u. Ze względu na potrzebę odpytywania również baz wiedzy w standardzie RDFS powstał również język RDQL (RDF DQL).

3.2.4 OKBC - Open Knowledge Base Connectivity

OKBC jest standardem API służącego do dostępu do systemów baz wiedzy. Standard ten jest niezależny od języka programowania, ani od konkretnego sposobu reprezentacji wiedzy w systemie. Zapewnia zwizły zbiór procedur służących głównie do odpytywania baz wiedzy. Jest zgodny między innymi ze standardem KIF.

3.2.5 OWL - Web Ontology Language

OWL powstał jako rozszerzenie standardu DAML+OIL i jest rozwijany w ramach projektu SemanticWeb. Zawiera on 3 podzbiory:

- **OWL Lite** - pozwala na budowanie struktur hierarchii, i stosowany powinien być tam gdzie potrzeba prostego języka reprezentacji danych
- **OWL DL** - umożliwia opisywania bardzo złożonych zbiorów danych, z tym jednak zastrzeżeniem aby był kompletny, odpowiada więc tak zwanej *Description Logic*
- **OWL Full** - umożliwia pełną dowolność syntaktyczną (prawie jak RDF), ale nie gwarantuje, że na podstawie takiej bazy wiedzy będzie można udowodnić każdą hipotezę

3.2.6 KIF - Format Wymiany Wiedzy

KIF (ang. *Knowledge Interchange Format*) jest standardem ANSI, który został stworzony aby opisywać rzeczy, zjawiska w sposób zrozumiały dla systemów przetwarzania informacji (baz danych, systemów ekspertowych, agentów programowych). Ponadto był on projektowany z myślą o zastosowaniu go jako "tłumacza" pomiędzy różnymi językami reprezentacji wiedzy. Język ten spełnia założenia języka rachunku predykatów pierwszego rzędu, dzięki czemu nie wpływa na poziom ontologiczny (operuje tylko w poziomie logicznym i epistemologicznym).

Język ten pozwala na przykład na zdefiniowanie konceptualizacji interesującej nas części rzeczywistości w sensie obiektów, funkcji i relacji.

3.2.7 Formal Concept Analysis

Jak pisze w [DP98], formalna ontologia jest ściśle związana z mereologią (relacje część-całość) oraz topologią (połączenia pomiędzy koncepcjami). W tym rozdziale przedstawimy najnowsze trendy związane z tymi pojęciami, które noszą nazwę: *formalna analiza konceptualna - FCA*. Jedyną jak na razie książką na ten temat jest [?].

Ogólnie FCA jest techniką pozwalającą na samo-uczenie i odkrywanie struktur konceptualnych (znaczeniowych) w danych. Te struktury są reprezentowane graficznie jako hierarchie konceptualne, co pozwala na łatwe stwierdzanie zależności w danych.

Podstawowym pojęciem w tej teorii jest pojęcie koncepcji (konceptu - ang. *concept*). Koncept składa się z dwóch części: ekstensji - czyli wszystkich obiektów należących do danego konceptu oraz z intensji - czyli zbioru wszystkich atrybutów wspólnych dla tych obiektów.

Drugim ważnym pojęciem jest kontekst czyli relacja określająca dokładnie jaki obiekt posiada jakie atrybuty (np. w postaci prostej tabeli relacyjnej bazy danych). Tak więc koncept określa nam jakie obiekty w danym kontekście posiadają pewien atrybut lub zbiór atrybutów. Oczywiście z punktu widzenia teorii te dwa pojęcia są równoważne, to znaczy mając dany kontekst możemy wyznaczyć wszystkie koncepty i na odwrót, mając daną siatkę konceptów możemy odtworzyć kontekst.

Tutaj pojawia się problem reprezentacji konceptów jako hierarchii określonej przez dany kontekst. Są one reprezentowane w postaci siatki konceptualnej (ang. *conceptual lattice*).

Przykład zastosowania formalnej analizy konceptualnej do zamodelowania prostej bazy wiedzy.

Założmy że mamy daną następującą tabelę np. z relacyjnej bazy danych opisującą popularne systemy operacyjne:

	darmowy	open-sou...	system op.	wsadowy	komercyjny	POSIX comp	real-time	wielozada...
Win 31			X	X	X			
Win 2000			X		X			X
DOS			X	X	X			
Linux	X	X	X			X		X
Solaris		X	X		X	X		X
OS/2			X		X			X
QNX		X	X		X	X	X	X
RT Linux	X	X	X			X	X	X
VxWorks			X		X		X	X

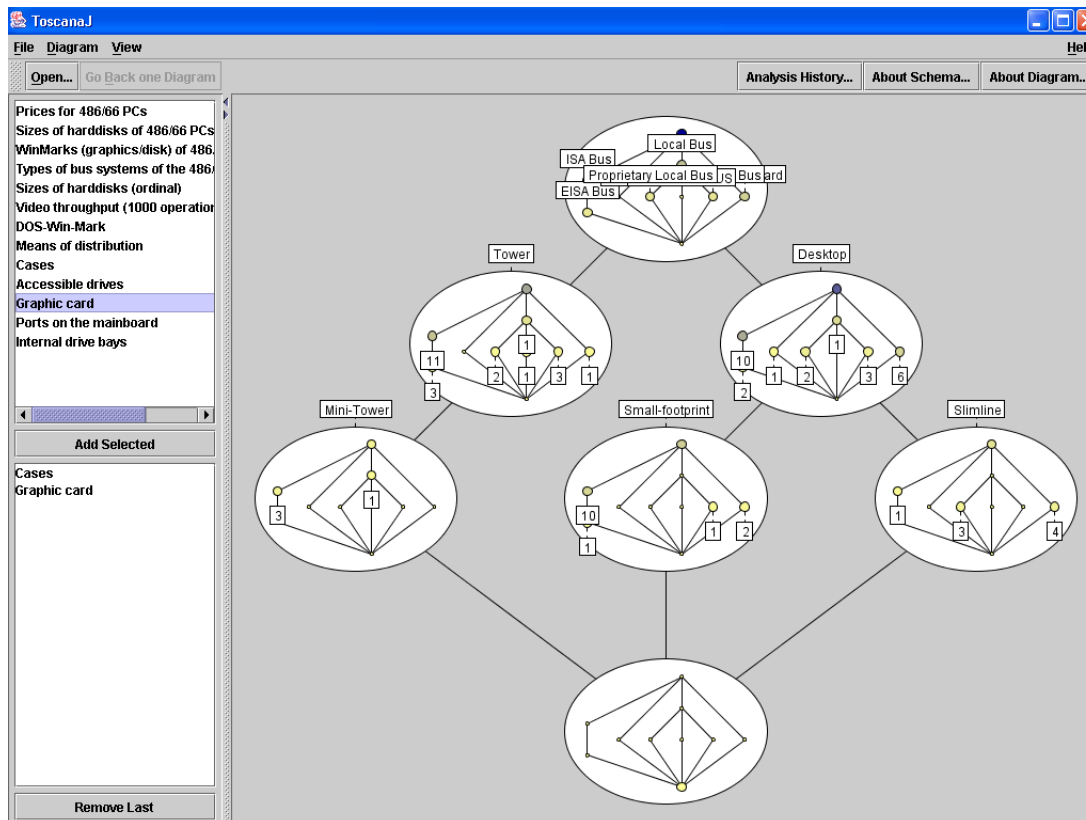
Rysunek 3: Tabela cech popularnych systemów operacyjnych.

W tym przypadku obiektami są konkretne systemy operacyjne, natomiast w kolumnach ustawione są cechy systemów operacyjnych, wraz cechą wspólną dla systemów operacyjnych czyli faktem że wszystkie są w istocie systemami operacyjnymi.

Tabela natomiast określa relację przypisującą konkretnym obiektom odpowiednie zbiory atrybutów. Poprzez zastosowanie metody FCA dostajemy następującą siatkę konceptualną:

Pojedyncze kółko na tej siatce przedstawia jeden koncept. Taki koncept jest więc zbiorem obiektów dzielących wspólnie pewien podzbiór zbioru atrybutów. Tak np. koncept oznaczony *Open-Source* posiada atrybuty { system operacyjny, wielozadaniowy, open-source}, i dzielą go obiekty {Linux, QNX, RTLinux}.

Przy takim przedstawieniu tej bazy wiedzy od razu widać jakie są zależności między konceptami. FCA jest szeroko stosowana do definiowania złożonych ontologii głównie z względu na bardzo zaawansowany aparat matematyczny podpirający tą teorię. Gotowe są tu przede wszystkim formalne algorytmy na zarządzanie ontologiami (uaktualnianie, łączenie itp.).



Rysunek 5: Przykład złożonej siatki konceptualnej w programie ToscanaJ.

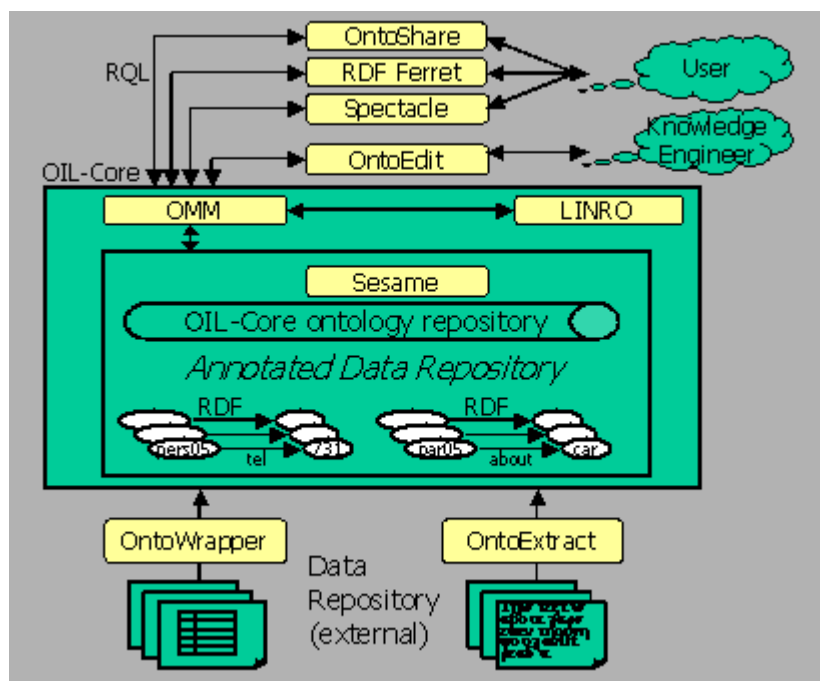
Ze względu na możliwość częstych zmian bazy wiedzy schemat bazy danych jest oparty na zmiennej liczbie tabel

- **KAON Api** - biblioteka umożliwiająca dostęp do bazy ontologii z poziomu języka programowania (na razie jest to Java), oraz umożliwia tworzenie własnych aplikacji opartych na tym środowisku
- **KAON Portal** - umożliwia tworzenie witryn WWW dających dostęp do baz ontologii zwykłym użytkownikom systemu
- **KAON Server** - centralny moduł środowiska. Łączy ze sobą pozostałe moduły
- **IO-Modeller** - narzędzie do tworzenia i uaktualniania ontologii w sposób wizualny
- **Text-To-Onto** - pozwala na pół-automatyczne tworzenie ontologii z tekstów źródłowych. Wykorzystuje do tego 3 algorytmy: *ekstrakcja terminów* (znajduje w tekście wyrazy potencjalnie nadające się na koncepty ontologii a człowiek później wybiera tylko te które rzeczywiście są interesujące), *ekstrakcja reguł* (algorytm wyszukuje potencjalne relacje pomiędzy konceptami wybranymi dla tego tekstu) oraz *oczyszczanie ontologii* (stosowany na koniec, wyszukuje koncepty, które potencjalnie nadają się do usunięcia z ontologii)

KAON jest wykorzystywany w wielu firmach oraz projektach w tym w Smart-Gov, projekcie Unii Europejskiej mającym na celu stworzenie systemu KM dla europejskiej biurokracji.

3.3.4 On-To-Knowledge

On-To-Knowledge (Ontology-based Tools for Knowledge Management) jest projektem realizowanym w ramach meta-projektu EU-IST. Głównym celem tego projektu jest stworzenie narzędzi i metod umożliwiających efektywne zarządzanie wiedzą w dużych organizacjach w oparciu o współdzielone bazy ontologii.



Rysunek 6: Struktura wewnętrzna systemu OntoKnowledge.

Ontologie w tym systemie służą takim głównym celom systemu KM jak:

- inteligentne wyszukiwanie zamiast wyszukiwania słów kluczowych
- odpowiadanie na pytania użytkowników
- możliwość zdefiniowania różnych *widoków* dla dokumentów

System umożliwia przeszukiwanie dokumentów tekstowych, wykorzystuje standardy RDF(S) oraz DAML+OIL do reprezentacji semantyki przechowywanych dokumentów oraz współdzielenie baz wiedzy.

Jego główne moduły to (patrz Rysunek 6):

- **OntoEdit** - umożliwia ręczne budowanie ontologii

- **OntoExtract** - pół-automatyczne tworzenie ontologii ze źródeł w języku naturalnym
- **OntoWrapper** - pół-automatyczne tworzenie ontologii z ustrukturyzowanych źródeł informacji
- **Sesame** - główne repozytorium, przechowuje wiedzę w formacie RDF oraz RDFS, może przechowywać dane w dowolnych bazach danych a nawet w zwykłych plikach.

3.3.5 Protege-2000

Protege-2000 jest narzędziem służącym do tworzenia systemów KM. Umożliwia on przede wszystkim wizualne modelowanie ontologii oraz tworzenie narzędzi do wydobycia informacji z baz ontologii. Główną zaletą tego systemu jest jego modularność pozwalająca na dowolne jego rozszerzanie poprzez tworzenie plug-in'ów. Obecnie dostępne są między innymi wtyczki umożliwiające analizę domeny przy użyciu FCA czy reprezentację ontologii w standardzie OWL.

4 Podsumowanie

Systemy KM czeka jeszcze długa droga zanim technologie rozwijane przez środowiska akademickie oraz te używane w wielkich przedsiębiorstwach będą miały wspólny mianownik. Należy mieć nadzieję, że takie projekty jak SemanticWeb upowszechnią formalne standardy reprezentacji wiedzy w środowiskach informatyków, co przełoży się na zainteresowanie nimi ze strony przemysłu. Tutaj obiecujący wydaje się fakt, że kilka nowoczesnych korporacji z dziedzin opartych na ścisłej wiedzy takie jak firmy farmaceutyczne i koncerny bio-technologiczne nie boją się korzystać z osiągnięć naukowców w dziedzinie reprezentacji wiedzy przy pomocy ontologii. Jednak prawdziwa popularność systemów formalnych będzie miała szansę nastąpić dopiero wtedy, gdy ktoś pokaże, że kierowanie wielkim przedsiębiorstwem może być dużo efektywniejsze przy użyciu formalnych metod reprezentacji wiedzy i wnioskowania.

A Zasoby w sieci

1. <http://www.primus.com>
2. <http://bprc.warwick.ac.uk/Kmweb.html#SEC1>
3. <http://meganesia.int.gu.edu.au/~phmartin/WebKB/doc/definitions.html#KnowledgeInWebKB>
4. <http://www.ontologging.com>
5. <http://kaon.semanticweb.org/>
6. <http://www.km-forum.org/>
7. <http://www.km.gov/>
8. http://www.kmlab.fiu.edu/index_.htm
9. <http://www.brint.com/km/>
10. <http://www.mbs.umd.edu/is/malavi/icis-97-KMS/>
11. <http://www.bus.utexas.edu/kman/>
12. <http://www.kmmag.co.uk/>
13. <http://www.kmnews.com/>
14. <http://www.cio.com/research/knowledge/>
15. <http://www.kmresource.com/>
16. <http://www.kvocentral.org/resources/km.html>
17. http://www.bitpipe.com/data/rlist?t=987097376_3716432&src=av_reports
18. <http://www.outsights.com/systems/kmgmt/kmgmt.htm#anex>
19. <http://www.insead.fr/CALT/Encyclopedia/ComputerSciences/Groupware/Workflow/>
20. <http://www-ksl.stanford.edu/researchthem.shtml>
21. <http://www.metakm.com/>
22. <http://www.proosto.megarolex.com/cms/go/pl/2>
23. <http://smok.pyton.pl/>
24. <http://ksi.cpsc.ucalgary.ca/KAW/KAW98/richards2/>
25. <http://meganesia.int.gu.edu.au/cgi-bin/projects/rentalFCA/BrowseREC.pl?context=central&map=y>
26. <http://meganesia.int.gu.edu.au/~phmartin/WebKB/kb/IT.html>
27. <http://www.sveiby.com.au/>
28. <http://www.totalkm.com/home.html>
29. <http://www.cyc.com/cyc-2-1/cover.html>

30. <http://toscanaj.sf.net>
31. <http://kaon.semanticweb.org>
32. <http://www.ontoknowledge.org>
33. <http://www.w3.org/RDF/>
34. <http://www.daml.org>
35. <http://www.daml.org/dql>
36. <http://www.ai.sri.com/~okbc>
37. <http://logic.stanford.edu/kif/kif.html>
38. <http://protege.stanford.edu>

Literatura

- [CE01] R. Cole and P. Eklund. *Browsing Semi-structured Web texts using Formal Concept Analysis*. Griffith Univ., Australia, 2001.
- [CM98] L. Chaudron and N. Maille. 1st order logic formal concept analysis: from logic programming to theory. *Linkoping Electronic Articles in Computer and Information Science*, 3(13), 1998.
- [dOBG00] A. de Oliveira Barroso and E. Pereira Gomes. *Knowledge Management: trying to understand*. Comissao Nacional de Energia Nuclear, 2000.
- [DP98] T. Davenport and L. Prusak. *Working Knowledge: How Organizations Manage What They Know*. Harvard Business School Press, 1998.
- [DRS02] J. Deogun, V. Raghavan, and H. Sever. *Association Mining and Formal Concept Analysis*. University of Nebraska, Lincoln NE, 2002.
- [Ein54] A. Einstein. *Ideas and opinions*. Dell, 1954.
- [GW99] B. Ganter and R. Wille. *Formal Concept Analysis. Mathematical Foundations*. Springer, 1999.
- [Hac98] R. D. Hackathorn. *Web farming for data warehouse*. Morgan-Kaufmann, 1998.
- [KC00] M. Kim and P. Compton. *Formal Concept Analysis for Domain-Specific Document Retrieval Systems*. Univ. of New South Wales, Sydney, 2000.
- [Lak87] G. Lakoff. *Women, Fire and Dangerous Things (What Categories Reveal About The Mind)*. University of Chicago Press, 1987.
- [Lee96] D. Leebaert, editor. *The Future Of Software*. MIT Press, 1996.
- [RB00] D. Richards and P. Bush. *Measuirng, Formalising and Modelling Tacit Knowledge*. Macquire Univ., Sydney, 2000.
- [RBD01] D. Richards, P. Bush, and C. Dampney. *Mapping tacit knowledge flows in organisation X*. Macquire Univ, Sydney, 2001.
- [S+00] W. Schorlemmer et al. *Formal Knowledge Management in distributed environments*. Univ. of Edinburgh, 2000.
- [Wil00] R. Wille. *Why can Concept Lattices support knowledge discovery in databases?* Technische Univ. Darmstadt, 2000.