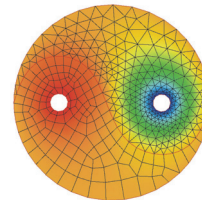




Publishing House  
AKAPIT



## **BAZA WIEDZY JAKO ELEMENT STERUJĄCY MODUŁEM INTERFEJSOWYM ZŁOŻONEGO SYSTEMU INFORMACYJNEGO**

**GRZEGORZ DOBROWOLSKI, EDWARD NAWARECKI**

Katedra Informatyki, Akademia Górniczo-Hutnicza, Kraków  
grzela@uci.agh.edu.pl

### **KNOWLEDGE BASE AS THE ELEMENT STEERING INTERFACE OF A COMPLEX INFORMATION SYSTEM**

#### **Abstract**

An idea of supporting a user facing a complex and functionally reach system under the assumption that his knowledge about it is incomplete and imprecise and can be expressed (and accumulated) by him using notions not agreed in meaning is presented in the paper. A description logics is proposed that describes the knowledge and constitute a functional basis of an interface of such system. A pilot version of the – built according to the logics – knowledge base and inference engine dedicated to INFOCAST Information System [Dobrowolski et al., 2003] is presented also.

**Key words:** knowledge-driven systems, human-computer interface, casting defects

### **1. WSTĘP**

Tradycyjne podejście do problematyki konstrukcji systemów informacyjnych zakłada określenie źródeł pochodzenia informacji oraz sposobów ich wykorzystania na etapie projektowania systemu. Zasoby informacyjne wykorzystywane przez tego typu systemy mają charakter dedykowany – informacje są gromadzone i przechowywane dla potrzeb danego systemu. Systemy mają charakter zamknięty, uzyskanie nowej funkcjonalności informacyjnej jest możliwe jedynie po załadowaniu nowych informacji (wiedzy, danych) do bazy systemu. Wykorzystanie sieci komputerowej ogranicza się do zapewnienia zdalnego dostępu do systemu, a w przypadku systemu rozproszonego do wymiany informacji pomiędzy poszczególnymi węzłami/modułami, przy czym struktura powiązań w systemie ma charakter sztywny, statyczny.

Dynamiczny rozwój globalnych sieci komputerowych spowodował, że możliwy jest dostęp do ogromnych zasobów informacyjnych o różnorodnej tematyce (przeznaczeniu) oraz strukturze (bazy danych, bazy wiedzy, systemy ekspertowe). Wywołuje to istotne zmiany w metodologii projektowania i realizacji systemów informacyjnych. W rozwiązaniach tego typu na całość systemu składa się zespół podsystemów powiązanych w pewną strukturę organizacyjną, powiązania między poszczególnymi elementami systemu są wynikiem charakteru posiadanej przez nie wiedzy i mogą być tworzone w sposób dynamiczny dla potrzeb realizacji konkretnego zadania.

Budowanie systemów tego typu wymaga nowych podejść i narzędzi do ich projektowania, programowania i uruchamiania (sprawdzania). Projektant takiego systemu staje obecnie przed koniecznością złożenia go głównie z istniejących (gotowych) elementów. Elementy te cechuje daleko posunięta heterogeniczność (człowiek musi być również roz-

ważany jako element systemu) oraz różnorodność pochodzenia, tak że sprawdzenie ich odpowiedniości jest konieczne; ponadto elementy te muszą być postrzegane o niespójności w zakresie reprezentowanej wiedzy.

Ogół problemów w tym zakresie daje się zamknąć pod hasłem konstrukcji systemu przy zachowaniu wspólnej ontologii – określonej, przyjętej konceptualizacji możliwie całej sfery pojęciowej związanej z systemem i jego funkcjonowaniem oraz formalnej reprezentacji tak określonej wiedzy [Gurino, 1998]. Wspólność oznacza tutaj, że ontologia ta ma być dzielona przez wszystkie moduły systemu, a ponadto przez odpowiednio zakreślone jego otoczenie, w tym użytkowników.

Jeżeli przyjąć, że ontologie poszczególnych modułów mogą być dostępne i użyte do budowania systemu, system taki nazywany będzie systemem *z wiedzą wyrażoną explicite* w przeciwieństwie do systemów tradycyjnych, w których co najmniej spora część *mądrości* jest wbudowana w strukturę i algorytmy, a więc w proponowanym podejściu – nie operacyjna. Wiedza *explicite* może stać się podstawą do projektowania (organizacji) lub nawet samoorganizacji takiego systemu.

Elementem systemu, który w szczególności sposób ogniskował będzie na sobie wskazane powyżej kwestie jest moduł interfejsowy. Problemy związane ze współdzieleniem ontologii pomiędzy tym modułem a użytkownikiem spotęgowane są założeniem, że komunikacja prowadzona być musi w oparciu o język naturalny. Problem powyższy został zidentyfikowany już z górami 20 lat temu, co znajduje wyraz w publikacjach, z których kilka przytoczyć można [Winograd, 1980; Lepratti i Berger, 2004].

W artykule skoncentrowano się na opracowaniu ogólnej konceptualizacji kwestii współdziałania człowieka ze złożonym i bogatym funkcjonalnie systemem przy założeniu, że wiedza człowieka o systemie jest niepełna i nieprecyzyjna oraz może być przez niego wyrażana i przyswajana za pośrednictwem terminów nie uzgodnionych znaczeniowo. Konceptualizacja ta jest rozwinięciem idei systemów z wiedzą wyrażoną *explicite*, w zakresie modułu interfejsowego [Dobrowolski i in., 2004].

Artykuł zredagowany jest następująco. Sekcja 2 zawiera dyskusję założeń funkcjonalnych takiego modułu interfejsowego. W sekcji następnej zaprezentowano logikę (deskrypcyjną [Baader i in., 2003; S. Staab i Studer, 2004]) przeznaczoną do opisu wiedzy, która stanowić ma podstawę funkcjonowania modułu – realizacji dialogu z użytkownikiem.

Sekcja 4 raportuje pokrótce pilotową wersję skonstruowanej odpowiednio do logiki bazy wiedzy i motoru wnioskowania, zilustrowaną przykładem stanowiącym odniesienie do fragmentu wiedzy systemu informacyjnego INFOCAST [Dobrowolski i in., 2003].

## 2. STEROWANY WIEDZĄ MODUŁ INTERFEJSOWY SYSTEMU

Ogólnym zadaniem omawianego modułu jest wspomaganie użytkownika w trakcie korzystania ze złożonego systemu informacyjnego jako wielofunkcyjnego narzędzia. Moduł interfejsowy systemu ma następujące zadania.

- Przechowywanie wiedzy o sferze użytkowej systemu, w tym jego możliwościach i właściwej procedurze zastosowania.
- Umożliwianie dialogu z użytkownikiem, zmierzającego do ustalenia właściwego trybu użycia, realizowanego poprzez zadawanie szeregu doprecyzowujących pytań. Stanowić ma to bazę do uczenia użytkownika systemu, w zasadzie bez konieczności odwołania się do klasycznego podrecznika.
- Prowadzenie dialogu bez założenia konieczności całkowitego współdzielenia terminologii (ontologii) dziedziny przedmiotowej systemu, jak również samego systemu.
- Umożliwienie rozbudowywania wiedzy o użytkowaniu systemu w trakcie jego pracy szczególnie w zakresie uwspólniania ontologii w kontakcie z nie tylko jednym użytkownikiem, systematyczna budowa możliwie szerokodostępnej ontologii. Mechanizm rozbudowy powinien być w maksymalnym stopniu zautomatyzowany, prowadzony również jako konsekwencja przebiegu dialogu roboczego.

Realizacja powyższych zadań powinna odbywać się przy założeniu minimalnego angażowania użytkownika.

Analiza powyższych założeń funkcjonalnych modułu prowadzi do oczywistego wniosku, że podstawowym problemem jego realizacji będzie wybranie odpowiedniej reprezentacji wiedzy a następnie obudowanie jej procedurami dowodzenia, protokołami interakcyjnymi aż wreszcie odpowiednio ekspresyjnym interfejsem użytkownika (tu w sensie GUI [Graphical User Interface]).

W dalszym ciągu artykułu zaproponowane zostanie rozwiązanie pierwszego z tych problemów, zaś raportowana implementacja eksperymentalna będzie miała za cel jedynie wsparcie analizy wła-



ściwości reprezentacji (odpowiedniej logiki opisowej) a nie prototypowanie całości modułu.

### 3. LOGIKA

#### 3.1. Elementy

Opis możliwych działań z systemem wykorzystywał będzie trzy kategorie pojęć:

*Act* – pojęcia odnoszące się do wykonywanych funkcji systemu, czynności użytkownika związanych z wykorzystaniem systemu oraz innych akcji sprawczych i ich abstrakcyjnych uogólnień.

*Mat* – pojęcia podmiotowo-przedmiotowe odpowiadające bytom materialnym związanym z ekstensjami pojęć *Act* oraz ich abstrakcyjne uogólnienia.

*Proc* – pojęcia proceduralne opisujące czynności, ale w kontekście wykonywania procedury właściwej dla systemu.

Pojęcia proceduralne stosowane będą do opisu możliwych trybów wykorzystania systemu, nie zaś określonych działań *Act* już przeprowadzonych w ramach realizacji jednej z procedur oferowanych przez system.

Scharakteryzowane powyżej pojęcia będą mogły być związane relacjami: równoważności (synonim), subsumcji, kompozycji. Właściwości logiczne tych relacji oraz ich semantyka, szczególnie w odniesieniu do subsumcji i kompozycji, zależą od kategorii wiązanych przez nie pojęć.

Oprócz relacji wprowadzonych powyżej dopuszcza się możliwość utrzymywania w ramach logiki bliższych opisów pojęć i relacji reprezentowanych przez ich atrybuty.

Z oczywistych względów relacja *synonim* (oznaczana  $\parallel$ ) może zachodzić tylko pomiędzy pojęciami tej samej kategorii. Jest relacją symetryczną, przechodnią i zwrotną.

Relacja *subsumcji* (oznaczana  $\triangleleft$ ) może wiązać dwa pojęcia podmiotowo-przedmiotowe *Mat*, dwa działania *Act* oraz działanie z pojęciem proceduralnym. Semantyka dwóch pierwszych przypadków jest naturalna (ogół-szczegół), natomiast w ostatnim z nich oznacza realizację działania w kontekście procedury. Jest relacją przechodnią, asymetryczną czyli relacją silnego porządku. Pomiędzy pojęciami subsumującymi z (dziedziczącymi po) danego pojęcia zachodzi alternatywa. W przypadku trzecim relacja posłużyć zatem może do opisanego alternatywnych sposobów realizacji procedury.

Relacje typu *kompozycji* (agregacji) (oznaczane  $\supset$ ) mogą zachodzić pomiędzy pojęciami różnych

kategorii i w zależności od tego będą posiadać różne właściwości. Poniżej przedyskutowane zostaną możliwe przypadki. Podstawową interpretacją znaczeniową pozostaje „całość-część”.

$Mat_1 \supset Mat_2$  – dla pary relacji tego typu zachodzi koniunkcja pomiędzy pojęciami agregowanymi przez to samo pojęcie.

$Act_1 \supset Act_2$  – dla pary relacji tego typu zachodzi alternatywa pomiędzy pojęciami agregowanymi przez to samo pojęcie. Aby opisać koniunkcję działań oraz – bardzo istotny przy tak postawionej domenie zastosowaniowej konstruowanej logiki – ciąg działań następujących po sobie, zastosowana będzie właściwość łączności dla relacji tego typu. Trzy sukcesywne działania zostaną zatem przedstawione w postaci  $Act_1 \supset Act_2 \supset Act_3$ . Aby objąć jednym pojęciem ciąg działań, należy posłużyć się pojęciem przedmiotowym będącym efektem takiego działania złożonego.

$Mat \supset Act$  – dla pary relacji tego typu zachodzi koniunkcja między pojęciami agregowanymi przez to samo pojęcie. Relacja interpretowana jest jako wskazanie warunku podmiotowo-przedmiotowego koniecznego do realizacji działania. W sytuacji, gdy opisać trzeba warunki alternatywne posłużyć się można pojęciem pomocniczym oraz relacją dziedziczenia.

$Act \supset Mat$  – dla pary relacji tego typu zachodzi alternatywa między pojęciami agregowanymi przez to samo pojęcie. Semantyka dla tego przypadku jest następująca – pojęcie *Mat* jest efektem pojęcia *Act*, wykonaniem działania.

$Proc_1 \supset Proc_2$  – dla pary relacji tego typu zachodzi alternatywa między pojęciami agregowanymi przez to samo pojęcie. Interpretacja dla tego przypadku to dwie alternatywne ścieżki przygotowujące do wykonania danego kroku w procedurze.

$Proc \supset Mat$  – dla pary relacji tego typu zachodzi alternatywa między pojęciami agregowanymi przez to samo pojęcie. Interpretacja to dwie różne procedury związane z tym samym pojęciem *Mat* (np. procedura tworzenia i procedura niszczenia).

$Proc \supset Act$  – dla pary relacji tego typu zachodzi alternatywa między pojęciami agregowanymi przez to samo pojęcie. Semantyka dla tego przypadku jest następująca – są to dwie różne procedury (np. procedura gotowania i procedura smażenia).

Bardziej skomplikowanym przypadkiem jest, gdy występuje agregacja kilku pojęć podmiotowo-przedmiotowych oraz kilku działań przez jedno działanie. Interpretacja takich przypadków jest następująca – wszystkie *Mat* są niezbędne i dlatego



zachodzi pomiędzy nimi koniunkcja, natomiast pomiędzy *Act* zachodzi alternatywa. Ponadto między każdym *Mat* a każdym *Act* zachodzi koniunkcja.

Innym przypadkiem złożonym jest agregacja poprzez pojęcie proceduralne. Interpretuje się to w ten sposób – agregacja przez procedurę oznacza, iż pojęcie agregowane istnieje tylko w momencie wykonywania procedury.

Stosując symbole wprowadzone powyżej dla relacji oraz odpowiednie piktogramy dla kategorii pojęć, szczególną conceptualizację działania pewnego systemu przedstawić można również w postaci grafu.

### 3.2. Procedury wnioskowania

Dla tak zdefiniowanej logiki zaprojektowano następujące procedury wnioskowania.

- Pytanie o pojęcie – wynikiem jest opis pojęcia (jego atrybuty), lista synonimów oraz lista bezpośrednich sąsiadów w sensie zdefiniowanych relacji. W przypadku pytania o pojęcie proceduralne za bezpośredniego sąsiada uważa się pojęcie, do którego należy dana procedura.
- Pytanie o dwa pojęcia – wynikiem jest podanie pojęć i relacji tworzących drogę w grafie, która łączy zadane pojęcia. Ponieważ zadanie takie w ogólnym przypadku nie ma jednego rozwiązania, omawiana procedura rozpada się na kilka podprocedur, różniących się sposobem konstrukcji przedmiotowej drogi. Możliwe jest tu poszukiwanie drogi najkrótszej, zgodnej z kierunkiem relacji subsumcji i kompozycji lub nie itp. W przypadku poszukiwania najkrótszej drogi powiązania poprzez pojęcie proceduralne traktuje się jako powiązanie bezpośrednie.
- Pytanie o procedurę w kontekście konkretnego pojęcia podmiotowo-przedmiotowego – odpowiedź są wszystkie alternatywne listy pojęć proceduralnych w odpowiedniej kolejności.

Dopuszczyć można również pytania o trzy i więcej pojęć. Przypadki takie da się przełożyć na kilka zapytań o dwa pojęcia. Muszą być przy tym odpowiednio uzgodnione sposoby konstrukcji odpowiednich ścieżek.

### 3.3. Semantyka zapytan

Przedstawione w punkcie poprzednim procedury wnioskowania mogą być następnie użyte do opracowania odpowiedzi na zapytania do bazy wiedzy skonstruowanej w oparciu o opisaną powyżej logikę.

Zakres znaczeniowy zapytań (i odpowiednio odpowiedzi) może być zakreślony następująco:

- Co oznacza dane pojęcie? – Podawany jest opis pojęcia i ewentualnie jego atrybuty.
- Jakie znamy inne określenia na to pojęcie? – Podawane są znane w bazie wiedzy synonimy danego pojęcia.
- Z czego składa się dane pojęcie? (Czego częścią jest dane pojęcie?) – Są to zapytania ogólne w sensie relacji kompozycji. Odpowiedziami mogą być: struktura wewnętrzna pojęcia podmiotowo-przedmiotowego, alternatywne sposoby realizacji działania, alternatywne ścieżki procedury, zestaw podmiotowo-przedmiotowy opisujący działanie, alternatywne procedury związane z pewnym pojęciem podmiotowo-przedmiotowym.
- Jakie znamy szczególne przypadki danego pojęcia? (Jakie znamy inne pojęcia danego rodzaju?) (Do jakiego rodzaju należy dane pojęcie?) Są to zapytania ogólne w sensie relacji subsumcji.
- W jaki sposób zrealizować dane pojęcie przedmiotowe? – Odpowiedź wskazuje odpowiednią procedurę.
- W jaki sposób wykonać dowolną procedurę związaną z danym pojęciem? – Odpowiedź wskazuje czynności składające się na daną procedurę.
- Czy istnieje (i jakie jest) pojęcie łączące daną grupę pojęć? – Ogólne pytanie na kojarzenie. Odpowiedź wyszukiwana w oparciu o poszukiwanie ścieżek w grafie bazy wiedzy.

Przeglądając powyższe możliwości, przyjąć należy, że baza wiedzy z tak określonym dostępem może stanowić pełne wsparcie dla nawet najbardziej skomplikowanych interakcji pomiędzy użytkownikiem systemu a inteligentnym komponentem interfejsowym.

## 4. PRZYKŁAD BAZY WIEDZY ORAZ JEJ FUNKCJONOWANIA

Poniżej zaprezentowano funkcjonowanie przykładowej bazy wiedzy skonstruowanej z wykorzystaniem zaproponowanej logiki. Rysunek 1 przedstawia graf fragmentu bazy wiedzy o systemie INFOCAST [Dobrowolski i in., 2003], która stanowi jednocześnie podstawę działania modułu interfejsowego. Baza ta opisuje realizację procedury diagnozy wady odlewu wraz z czynnościami, jakie musi wykonać zarówno użytkownik jak i system (ustawionymi względem siebie w czasie) oraz uwarunkowania podmiotowo-przedmiotowe ich realizacji. Wśród







bezpośrednio widoczny dla użytkownika, pozostając przykrytym przez właściwy interfejs i protokoły komunikacyjne.

## 5. PODSUMOWANIE

W artykule przedstawiono koncepcję budowy modułu interfejsowego złożonego systemu informacyjnego. Punktem wyjścia dla tej koncepcji jest zapisanie pełnej wiedzy o aspekcie użytkowym (funkcjach) systemu w formie pozwalającej na automatyczne przetwarzanie. Zaproponowano logikę (deskrypcyjną) w zastosowaniu do opisu wiedzy oraz pilotową wersję skonstruowanej odpowiednio do tej logiki bazy wiedzy i motoru wnioskowania.

Testy przeprowadzone dla prototypu wykazały spełnienie podstawowych założeń i celów. Wyniki tych testów pozwoliły sformułować następujące wnioski dotyczące implementacji docelowej.

Pomimo wielu zalet systemu JTP wydaje się on dla celów budowy modułu interfejsowego zbyt rozbudowany. Dobrze zdefiniowana logika bazy wiedzy, jak również niezbyt złożone wymagania co do mechanizmu wnioskującego skłaniają do implementacji w oparciu o własne rozwiązania. Zastosować tu można składowanie bazy wiedzy w relacyjnej bazie danych o odpowiedniej strukturze oraz wykorzystanie motoru bazy danych jako podstawy mechanizmu wnioskującego. Taka konstrukcja okaże się również korzystna dla realizacji funkcji modyfikacji bazy wiedzy.

Na zakończenie należy podkreślić, że opisany moduł interfejsowy dobudowany być może do praktycznie dowolnego systemu, w tym również wspomagającego procesy produkcyjne w przemyśle przetwórstwa metali.

## LITERATURA

- Baader, F., McGuinness, D. L., Nardi, D., Patel-Schneider, P.F., editors (2003). *THE DESCRIPTION LOGIC HANDBOOK: Theory, implementation, and applications*. Cambridge University Press.
- Dobrowolski, G., Kisiel-Dorohinicki, M., Nawarecki, E. (2004). Agent technology in information systems with explicit knowledge. In Bubnicki, Z. and Grzech, A., editors, *Proc. of the 15th Int. Conf. on Systems Science*, volume III, pages 105-112. Oficyna Wydawnicza Politechniki Wrocławskiej, Poland.
- Dobrowolski, G., Marcjan, R., Nawarecki, E., Kluska-Nawarecka, S., Dziadus, J. (2003). Development of INFOCAST: Information system for foundry industry. *TASK Quarterly*, 7(2):283-289.
- Fikes, R., Frank, G., Jenkins, J. (2002). JTP: An object-oriented modular reasoning system. <http://www.ksl.stanford.edu/software/JTP/>.

- Guarino, N. (1998). Formal ontology and information systems. In Guarino, N., editor, *Formal Ontology in Information Systems. Proceedings of FOIS'98*, pages 3-15, Amsterdam. IOS Press.
- Lepratti, L., Berger, U. (2004). Ontology-based approach for the future production of European Industry. In Lefranc, G., editor, *Management and Control of Production and Logistics 2004*, pages 123-128, Santiago de Chile. IFAC/IEEE/ACCA, Universidad de Las Américas.
- Staab, S., Studer, R., editors (2004). *Handbook on Ontologies*. Springer.
- Winograd, T. (1980). What does it mean to understand language? *Cognitive Science*, (4):209-241.

Received: June 26, 2006  
Received in a revised form: September 19, 2006  
Accepted: September 21, 2006

