
Reprezentacja wiedzy dla rekonfigurowalnych systemów automatyzacji*

Ola Angelsmark¹, Jacek Malec¹, Klas Nilsson¹, Sławomir Nowaczyk¹,
Leonardo Prospero¹

Streszczenie

Artykuł opisuje prace prowadzone nad formalizmami reprezentacji wiedzy przydatnymi w europejskim projekcie SIARAS. *Inspekcja i montaż oparte na umiejętnościach, dla rekonfigurowalnych systemów automatyzacji* jest projektem mającym na celu stworzenie inteligentnego systemu wspomagającego rekonfigurację i adaptację systemów produkcyjnych. Wiedza jest reprezentowana w strukturze nazywanej *ontologią*, wyrażonej w języku OWL dla celów ogólnych rozumowań w logice deskryptywnej, oraz w pewnej liczbie specyficznych modułów rozumujących dopasowanych do dziedziny zastosowania systemu.

1. WSTĘP

Reprezentacja wiedzy jest jedną z najintensywniej studiowanych dziedzin w całej 50-letniej historii sztucznej inteligencji. W tym okresie zaproponowano setki metod, poczynając od podejść formalnych, opartych na językach logiki, poprzez metody półformalne, a kończąc na systemach budowanych ad-hoc, stosowanych tylko w pojedynczych przypadkach.

Reprezentacja wiedzy, podobnie jak i inne dziedziny sztucznej inteligencji, podlegała zmianie paradygmatów a nawet, w pewnym stopniu, w sposób ciągły adaptowała się do zmian w upodobaniach sponsorów badań naukowych. Ostatnie dziesięciolecie było n.p. poświęcone w dużej mierze systemom wieloagentowym i rozwojowi tzw. *sieci semantycznej*. Projekt sieci semantycznej [10] ma na celu sformalizowanie dużych fragmentów wiedzy w takiej postaci, aby zapewnić wspólne zrozumienie podstawowych pojęć w systemach rozproszonych i w ten sposób zagwarantować ich bezpieczną współpracę. Pojęcie *ontologia* jest zwykle używane w tym kontekście do opisu logicznej formalizacji (korzystającej z jednego z dostępnych języków logiki) konkretnej (pod)dziedziny wiedzy, zeskładowanej w powszechnie zrozumiałym formacie i dostępnej poprzez światową sieć WWW bądź inny temu podobny mechanizm.

Obszar zastosowań tych badań nie ogranicza się do sieci WWW. Poniżej przedstawiamy wstępne prace w projekcie mającym na celu wprowadzenie ograni-

*Praca finansowana przez projekt SIARAS (FP6 - 017146).

¹Department of Computer Science, Lund University, Box 118, S-221 00 Lund, Sweden, {olaan,jacek,klas,slawek}@cs.lth.se.

czonych elementów sztucznej inteligencji do planowania i nadzorowania produkcji (fazy projektowania i integracji systemu). Zasadniczym celem projektu jest wspomaganie inżynierów i ułatwianie (prze)projektowywania produkcji, a w ten sposób obniżanie kosztów w większości typowych zastosowań. Projekt SIARAS jest finansowany przez Unię Europejską w ramach 6-go programu ramowego w nadziei, że uzyskane rezultaty i produkty pozwolą wesprzeć wiedzę i produktywność w małych i średnich przedsiębiorstwach; jednak ten dalekosiężny cel pozostaje poza zakresem tego artykułu.

Artykuł opisuje krótko projekt, rozważa możliwości doboru sposobów reprezentacji wiedzy, opisuje przyjęte metody, dyskutuje konsekwencje tego doboru, a następnie przedstawia powiązane badania i pewne wstępne konkluzje z już wykonanych prac.

2. SIARAS

SIARAS jest akronimem projektu typu STREP finansowanego przez Unię Europejską (FP6 - 017146) i zatytułowanego *Inspekcja i montaż oparte na umiejętnościach, dla rekonfigurowalnych systemów automatyzacji* (ang. Skill-based Inspection and Assembly for Reconfigurable Automation Systems). Jego zasadniczym celem jest budowa inteligentnego systemu, wstępnie nazwanego *Serwerem Umiejętności*, zdolnego wspomóc automatyczną i semi-automatyczną rekonfigurację procesu produkcyjnego.

Głównym celem prac nad specyfikacją systemu było uwzględnienie dwóch, częściowo sprzecznych ze sobą podejść do procesu rekonfiguracji: wywodzącego się ze sztucznej inteligencji *top-down* z góry do dołu i inżynierskiego *bottom-up* z dołu do góry.

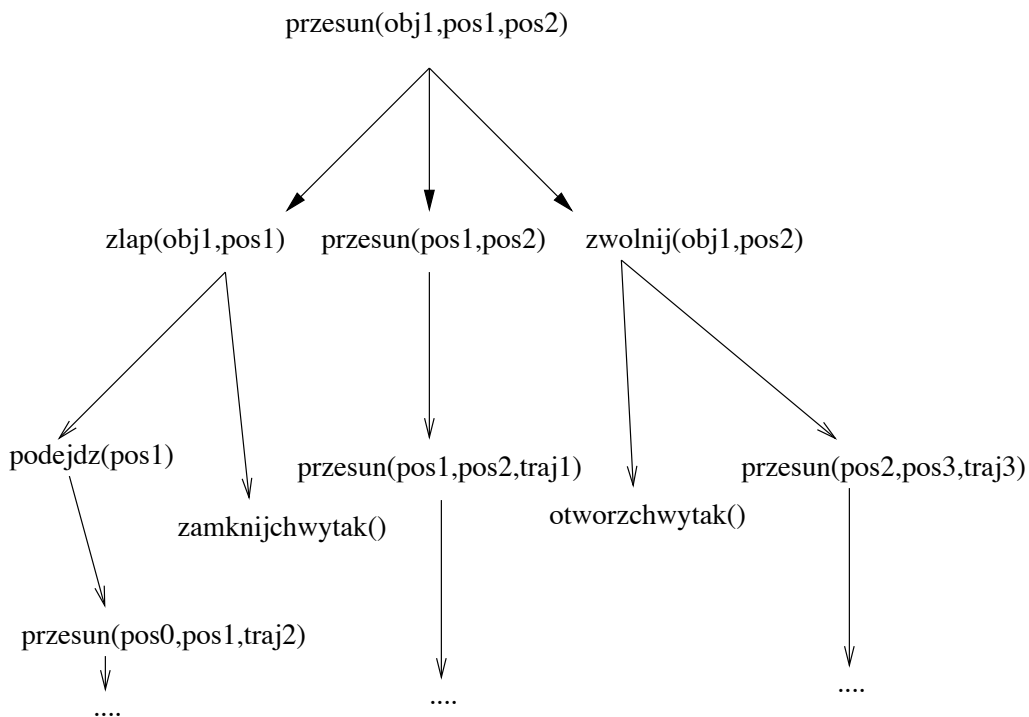
2.1. Podejście „z góry do dołu”

Podejście „z góry do dołu”, typowe dla klasycznej Sztucznej Inteligencji, polega na opisanu problemu rekonfiguracji przy użyciu terminologii typowej dla planowania: mając dane nowe zadanie (zwykle wyrażone w postaci *celu* do osiągnięcia), być może będące modyfikacją poprzedniego, oraz zbiór umiejętności dostępnych w systemie — rozumianych jako opisy operacji które mogą zostać wykonane przez dostępne urządzenia — znaleźć sekwencję tych operacji gwarantującą prawidłowe wykonanie zadania (innymi słowy, znaleźć plan po wykonaniu którego cel zostanie osiągnięty).

Terminy takie jak *zadanie*, *umiejętność*, *operacja* czy *plan* wymagają formalnych definicji. W SZI są one zwykle reprezentowane w sposób symboliczny, formalny i mocno abstrakcyjny. Typowym przykładem byłoby stwierdzenie, że robot może wykonać operację **przesuń(obiekt1,z_poz,na_poz)** w wyniku czego sytuacja zmieni się z takiej w której **w(obiekt1,z_poz)** na taką w której **w(obiekt1,na_poz)**. Taki poziom abstrakcji jest oczywiście mało interesujący dla inżyniera zajmującego się produkcją, dlatego też łatwo sobie wyobrazić, że konieczne będzie zdefiniowanie odpowiedniej hierarchii poziomów abstrakcji. Prowadzi to do definicji *planowania hierarchicznego*, w którym abstrakcyjne operacje są kolejno coraz bardziej ukonkretniane, często wielokrotnie, zanim powstanie ostateczny, realizowalny plan. W po-

wyższym, bardzo prostym przykładzie można wyodrębnić co najmniej trzy poziomy abstrakcji (por. rys. 1):

- **przesun(obj1,pos1,pos2)**
- **przesun(pos1,pos2)** (być może trzymając jakiś obiekt)
- Przesun manipulator robota z pozycji $(x,y,z,\alpha,\beta,\gamma)$ do nowej pozycji $(x',y',z',\alpha',\beta',\gamma')$ (globalne współrzędne kartezjańskie / biegunowe)
- Przesun manipulator robota z pozycji $(x,y,z,\alpha,\beta,\gamma)$ do nowej pozycji $(x',y',z',\alpha',\beta',\gamma')$ poruszając się po bezpiecznej trajektorii $f(x,y,z,t)$
- Przesun manipulator robota z pozycji $(x,y,z,\alpha,\beta,\gamma)$ do nowej pozycji $(x',y',z',\alpha',\beta',\gamma')$ poruszając się po bezpiecznej trajektorii $f(x,y,z,t)$ i minimalizując zużycie energii (minimalizując czas, itp.)



Rys. 1. Fragment hierarchii poziomów abstrakcji

Serwer Umiejętności musi zawierać odpowiednie reprezentacje dla każdego z tych poziomów i na każdym z nich musi być w stanie przeprowadzać odpowiednie obliczenia. W tym celu system wymaga zdefiniowania odpowiedniej hierarchii formalizmów. Powinna ona zawierać na poziomach abstrakcyjnych formalizmy symboliczne takie jak logiki, oraz modele kinematyczne i dynamiczne, planowanie ścieżek bezkolizyjnych i równania różniczkowe na poziomach bardziej konkretnych.

Jedną z technik wyjątkowo dobrze nadających się do tego celu, a dodatkowo oferującą adekwatne narzędzia obliczeniowe, jest *Hierarchiczne planowanie sieci zadań* (ang. Hierarchical Task Network Planning, por. Rozdział 11 w [4]). Wiele nowo-

czesnych systemów automatycznego planowania ma bardzo efektywne implementacje które doskonale radzą sobie z ograniczonymi dziedzinami, takimi jak te będące przedmiotem zainteresowań projektu SIARAS.

W dalszej części tego artykułu wprowadzimy dwie hierarchie opisujące wiedzę na temat *umiejętności* (tzn. możliwości oferowanych faktycznie lub potencjalnie przez urządzenia) oraz *zadań* (tzn. ogólnej reprezentacji działań, których użytkownik może zażądać od systemu). Na najwyższym poziomie zadaniem Serwera Umiejętności (dalej: SU) może być zidentyfikowanie dostępnych w rozważanym systemie umiejętności, zidentyfikowanie zadania do wykonania (opisanego w jakiś sposób przez użytkownika) oraz stworzenie planu, który doprowadzi do osiągnięcia celu. Oznacza to, że dwie powyższe hierarchie muszą się w jakimś punkcie spotkać, co odpowiada konkretnemu, realizowalnemu planowi w którym każde pojedyncze zadanie elementarne zostaje zrealizowane przez konkretną umiejętność dostępną w systemie.

Zagwarantowanie takiego wspólnego punktu obydwu hierarchii pozwoli nam wnioskować o parametrach konkretnych zadań (akcji w terminologii SzI, operacji w robotyce), aż do poziomu konkretnych programów sterujących urządzeniami, jeśli okaże się to konieczne. Zadanie uzyska konkretne wartości parametrów albo z opisu umiejętności (na poziomie symbolicznym) albo z dołączonej bazy danych programów sterujących, tzn. z konkretnej realizacji danej umiejętności oferowanej przez konkretne urządzenie.

Jednym z interesujących aspektów związanych z tym podejściem jest konieczność rozważania czujników jako urządzeń oferujących umiejętności *zbierania wiedzy*, tzn. takich które tworzą nowe dane wejściowe dla dalszych etapów procesu produkcji. Wykorzystany system automatycznego planowania musi zatem pozwalać na modelowanie akcji „tworzących wiedzę” — co jest samo w sobie problemem trudnym. Tym niemniej, nie będziemy tego zagadnienia szerzej omawiać w tym artykule.

Zasadniczo, każde urządzenie może być opisane na wiele sposobów, zależnie od wymogów konkretnego zastosowania. Na przykład, kamera CCD może być używana do wykrywania ruchu, koloru lub obecności, jako urządzenie przeprowadzające kontrolę jakości (produkt dobry lub wadliwy), klasyfikujące, jako urządzenie oferujące sprzężenie zwrotne (podczas np. spawania lub cięcia), itp. Na najniższym poziomie kamera może być opisana poprzez swoje parametry fizyczne, takie jak liczba pikseli, czułość, geometria soczewek, pole widzenia, itd. Na najwyższym poziomie, kamera odpowiada dokładnie jednemu z zadań zdefiniowanych przez użytkownika. Łatwo wyobrazić sobie wiele poziomów pośrednich, których szczegółowa implementacja zależy od tego na ile SU powinien być w stanie wnioskować o urządzeniach i ich umiejętnościach.

Innym interesującym problemem jest sposób opisywania agregacji sensorycznej (ang. sensor fusion), tzn. jak wyrazić i wnioskować o tym, że pewna liczba prostych czujników odległości może być (jako całość) użyta do triangulacji trójwymiarowej. Czy tego typu umiejętności *złożone* powinny być opisane jako operacje abstrakcyjne, z wyspecyfikowanymi warunkami pre- i post-? Wydaje się, że wiedza niezbędna do wnioskowania na tym poziomie jest zbyt rozległa i skomplikowana, żeby dało się ją sformalizować w ogólnej postaci. Zakładamy, że w sytuacjach kiedy będzie to niezbędne, złożone operacje tego typu będą opisane jako całość — w celu ograniczenia złożoności.

Podsumowując, SU wymaga potężnej bazy wiedzy, składającej się z wielu różnorodnych reprezentacji zarówno umiejętności jak i zadań. Dodatkowo konieczny jest opis konkretnego procesu produkcyjnego (w tym opis ostatecznego produktu, na przykład m.in. w postaci modelu CAD), bazy danych o dostępnych urządzeniach, oraz wydajnego systemu planowania. Wynikiem działania SU będzie specyfikacja procesu produkcji spełniająca założenia zdefiniowane przez użytkownika, lub informacja dlaczego wymagania nie mogły zostać spełnione i sugerowane zmiany, które pozwoliłyby uniknąć problemu.

W ogólnym przypadku złożoność takiego rozwiązania byłaby oczywiście nie do przyjęcia. Jednakże w ograniczonej dziedzinie, z jaką mamy do czynienia w ramach projektu SIARAS, istnieją szanse na rozwiązanie które byłoby obliczeniowo wykonalne. Na tym etapie zaawansowania projektu wydaje się, że sformalizowanie wiedzy jest najpoważniejszym problemem.

2.2. Podejście reparametryzacyjne „z dołu do góry”

Założeniem przy tym podejściu jest, że Serwer Umiejętności (SU) jest wykorzystywany wyłącznie do rekonfiguracji istniejącego, poprawnego i właściwie opisanego procesu produkcji. Nie oczekuje się od systemu, że zaproponuje nowe rozwiązania, czy że będzie poszukiwał alternatywnych sposobów realizacji procesu. Przeciwnie, dostarczony opis powinien być kompletny, a proces wykonywalny i poprawnie sparametryzowany. W szczególności, powinien istnieć opis zadania, tzn. co jest produkowane, jak (jakie są kroki, czy też stany procesu, tzn. podzadania) i w jakim stopniu każdy krok (podzadanie) dokłada się do osiągnięcia celu. W konkretnym przypadku powinna to być struktura drzewiasta. Baza wiedzy będzie zawierała pewną ilość takich drzew, być może reprezentowaną przez zorientowany graf acykliczny. Język specyfikacji pozostawiamy tutaj nieokreślony.

W tym podejściu umiejętności opisują każde urządzenie jako zdolne do realizacji pewnych (pod)zadań — a zatem przypisanie do liści grafu zadań powinno być trywialne. Interesujące wydają się tutaj problemy związane ze specyfikacją warunków granicznych pomiędzy zadaniami elementarnymi (rozdzielenie sukces/porażka, warunki łączenia dwóch zadań równoległe lub szeregowe, itd.) i, ponownie, hierarchia opisów umiejętności, od abstrakcyjnych do konkretnych.

SU powinien zajmować się tworzeniem, przechowywaniem i analizowaniem wiedzy o przypisaniu umiejętności zadaniom, o czasowej zależności pomiędzy zadaniami oraz o szczegółach realizacji umiejętności (sparametryzowany opis, może nawet program sterujący urządzeniem). Zapytanie do SU odbywać się będzie w następujący sposób: inżynier (odpowiednio przygotowany użytkownik) modyfikuje opis procesu w jakiś sposób. Modyfikacja ta może polegać na zmianie jakiegoś parametru, wymianie podzadania na inne, zmianie warunków brzegowych, itp. Przedmiotem dyskusji jest interfejs użytkownika, ponieważ w sposób zasadniczy będzie on wpływał na repertuar możliwych pytań.

SU powinien spełniać następujące funkcje:

1. sprawdzenie poprawności opisu procesu (składnia, parametry);
2. modyfikacja aktualnej implementacji procesu; albo reparametryzacja procesu, bądź jego rekonfiguracja;

3. opis napotkanych problemów;
4. opis możliwych sposobów rozwiązania problemów;
5. sugestia rozwiązania (zależnie od słownika).

W kroku 2, po przyjęciu i sprawdzeniu żądania użytkownika, SU powinien być w stanie zanalizować nową wersję procesu biorąc pod uwagę następujące własności:

- poprawność (czy urządzenia posiadają wszystkie umiejętności wymagane przez nowe zadanie);
- dokładność;
- powtarzalność, giętkość, niezawodność;
- czas, zużycie energii;
- bezkolizyjność;
- zdolność unikania i analizy błędów;
- koszt.

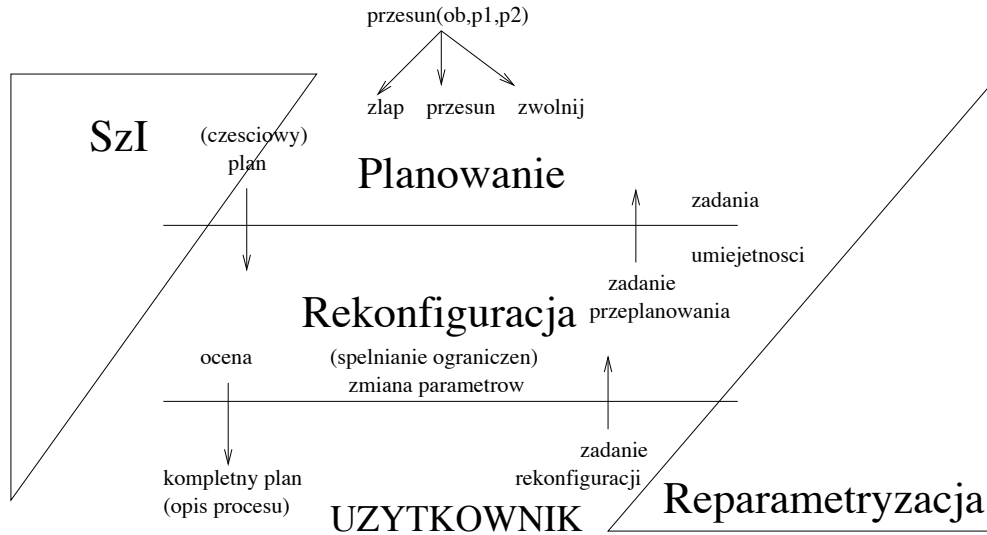
Kroki 4 i 5 muszą być powtarzane dopóki użytkownik ostatecznie nie zaakceptuje rozwiązania, bądź dopóki system nie dojdzie do wniosku, że więcej rozwiązań nie będzie w stanie dostarczyć. Zasadniczym celem systemu jest **obniżenie kosztu i czasu reparametryzacji bądź rekonfiguracji procesów produkcyjnych**. W szczególności nie należy oczekiwać, że SU będzie w stanie zaprojektować nowy system produkcyjny od podstaw. Celem projektu jest zbudowanie systemu, który będzie się wyróżniał łatwością obsługi i rozszerzonym wspomaganie użytkownika, porównując z innymi znanymi systemami tego rodzaju.

2.3. Poszukiwanie złotego środka

Wydaje się, że podejście sztucznej inteligencji, z góry do dołu, jest obliczeniowo zbyt złożone poza bardzo ograniczonymi przypadkami, podczas gdy podejściu reparametryzacyjnemu brakuje ogólności, ponieważ musiałoby ono w zasadzie polegać na utrzymywaniu bazy danych możliwych ustawień parametrów dla wielu urządzeń i na wyszukiwaniu ich (być może uzupełniając to symulacją bądź specjalizowanymi algorytmami optymalizacji) w chwili otrzymania właściwego zapytania. Podstawowym kłopotem może też być skalowalność i rozszerzalność systemu. Istnieją uzasadnione obawy, że przy tym podejściu system będzie ograniczony tylko do pewnej wąskiej grupy zastosowań, bez naturalnych możliwości rozbudowy. Z drugiej strony, rozwiązanie oparte na bazie danych zawierającej szczegóły dotyczące urządzeń będzie *nieuniknione* w każdym sensownym rozwiązaniu problemu.

Aby powiązać te dwa punkty widzenia proponujemy rozwiązanie wielopoziomowe, które pozwoli z jednej strony na rekonfigurację opartą o dane szczegółowe, o ile to tylko możliwe, ale przewidujące przeplanowywanie działań na wyższym poziomie, jeśli okaże się to konieczne, i tylko wtedy (por. rys. 2).

Przy podejściu wielopoziomowym jednym z podstawowych problemów jest podział żądania użytkownika na rekonfigurację i przeplanowanie, tzn. jaka część żądania może być rozwiązana poprzez analizę ograniczeń w ramach rekonfiguracji, a jaka wymaga ponownego planowania. Częściowo jest to związane ze złożonością rozumowania wymaganego do odpowiedzi na pytanie: w zależności od tego, ile jawnej wiedzy oczekujemy od inżyniera opisującego proces a jak dużo będzie należało oprócz niej wydedukować. Ten problem jest ściśle związany z zadaniem *rozpoznawania pla-*



Rys. 2. Poziomy funkcjonalne

nów (por. [4]), gdy automatyczny system rozumujący ma za zadanie odkryć cel konkretnego ciągu działań.

Kolejnym interesującym wyzwaniem jest dołączenie wielu szczegółowych algorytmów specyficznych dla poszczególnych umiejętności (jak np. planowanie bezkolizyjnej ścieżki ruchu robota, symulacje, analiza modeli algebraicznych, itd.) i zapewnienie, żeby Serwer Umiejętności je zauważał, o nich rozumował i z nich korzystał w sposób najbardziej efektywny.

3. ONTOLOGIA

Zdecydowaliśmy oprzeć reprezentację wiedzy w systemie na pojęciach *urządzenie* (obiekty fizyczne dostarczane przez swoich producentów) i *umiejętność*, podczas gdy *zadanie* i jego opis istnieją tylko dynamicznie, w trakcie sesji użytkownika z Serwerem Umiejętności. Zadania mogą być uważane za (nb. dość złożone) kombinacje umiejętności i z tego powodu nie muszą być wyróżniane jako samoistny element słownika.

Statyczna część wiedzy jest reprezentowana w *ontologii*: strukturze danych zawierającej wszystkie istotne relacje pomiędzy używanymi pojęciami. Dość często ontologie są używane do celów klasyfikacji. W przypadku Serwera Umiejętności klasyfikacja dokonywana jest w chwili rejestrowania nowych urządzeń w systemie, zatem możemy ontologię traktować jako czystą taksonomię. Ontologia SU jest systemem rozproszonym, zawierającym dość złożoną strukturę umiejętności oraz biblioteki urządzeń będących liśćmi drzewa klasyfikacji.

Uważnie dobrany zbiór pierwotnych pojęć reprezentacji, łącznie z dość bo-

gata ontologią i z zestawem efektywnych (tzn. wielomianowych) algorytmów rozumujących (istniejących dla pewnych klas logik deskryptywnych) pozwalają nam pozostawić otwartą możliwość rozszerzania systemu, jednocześnie skupiając się na konkretnych przypadkach ilustrujących ideę Serwera Umiejętności. Ponieważ jednym z partnerów projektu jest grecko-niemieckie przedsiębiorstwo INOS, zajmujące się integracją systemów sensorycznych w przemyśle samochodowym, skupiamy się na kilku przypadkach zasugerowanych przez tego partnera.

Jako narzędzie tworzenia ontologii i początkowej nią manipulacji wybraliśmy narzędzie otwartego źródła Protégé [9], ze względu na jego otwartość i modyfikowalność. W szczególności, Protégé pozwala na dobór systemu wnioskującego do złożoności użytej reprezentacji, posiada elastyczny interfejs do kilku języków programowania i systemów baz danych oraz oferuje wiele sposobów wizualizacji reprezentowanej wiedzy.

4. FORMALIZMY

W przyjętym podejściu ontologia jest używana do rozumowania o umiejętnościach odpowiednich do wykonywania zadań (po pewnej wstępnej parametryzacji) i o urządzeniach posiadających i oferujących te umiejętności (w pewnych warunkach). Czysta ontologia mogłaby być używana do wyszukiwania, dopasowywania wzorców i prostej klasyfikacji, podczas gdy inne rodzaje rozumowania, jak planowanie, optymalizacja, stwierdzanie spójności, itp., wymagają bardziej zaawansowanych narzędzi rozumujących, zarówno ogólnych jak i dostosowanych do konkretnych aplikacji. Narzędziami szerokiego stosowania które dotychczas wykorzystywaliśmy w Serwerze Umiejętności są Racer [5], Fact++[6], Algernon i Pellet. Różnią się one mocą obliczeniową i efektywnością, będąc zdolne do analizy ontologii wyrażonej w ograniczonym języku logiki deskryptywnej [8] (jak np. OWL-DL w systemie Protégé) w sposób efektywny (wielomianowo) [2], bądź też analizujące bardziej ekspresywny pełny język OWL [11], ale kosztem eksponencyjnych algorytmów rozumowania. Istnieje też możliwość doboru algorytmu rozumującego do zadanego pytania, co daje jeszcze większe zdolności adaptacyjne systemu.

W ramach projektu tworzone są specjalne narzędzia do uzupełniania i wyszukiwania wiedzy w strukturach ontologii, co z jednej strony pozwoli na rozszerzenie systemu przez dostawców technologii, a z drugiej pozwoli na utrzymywanie systemu w postaci rozproszonej, w szczególności biblioteki urządzeń mogłyby być przechowywane u producentów urządzeń i udostępniane poprzez WWW. Rozważany jest także protokół specyfikacji rozumowań i obliczeń niezbędnych do znalezienia odpowiedzi na pytania użytkownika. Zarówno systemy ogólnego stosowania, jak i rozwiązania specyficzne dla konkretnej aplikacji muszą być widziane i używane przez SU w dokładnie ten sam sposób, co wymusza jasny protokół dostępu do modułów rozumujących.

5. PRACE ZWIĄZANE

Różnorodność badań nad reprezentacją wiedzy została udokumentowana w wielu pracach, zarówno w podstawowych podręcznikach sztucznej inteligencji, jak i w bardziej specjalizowanych źródłach i czasopismach naukowych. Jedną z najnowszych pozycji wyczerpująco prezentujących tę dziedzinę jest [1].

Organizacja która upowszechniła pojęcie sieci semantycznej, a w szczególności pojęcie *ontologii*, Konsorcjum W3C, udostępnia wszystkie dokumenty na swoim serwerze sieciowym [10]. W szczególności, specyfikacje najpopularniejszych obecnie formalizmów reprezentacji wiedzy, jak OWL [11] czy DAML-OIL [3], łącznie z odnośnikami do narzędzi przetwarzających wiedzę w tych językach, są tam dostępne.

Planowanie produkcji jest zwykle uważane (w sztucznej inteligencji) za poddziedzinę planowania automatycznego. Jednak oprócz klasycznych wyników analizy wytwarzalności, przedstawionych ostatnio w [4], nie ma w zasadzie publikacji dotyczących użycia ontologii w automatycznym planowaniu produkcji. Niemniej jednak, istnieje bogata literatura dotycząca wspomaganie działalności inżynierskiej w planowaniu produkcji poprzez szczegółowe modelowanie procesów i rozwój narzędzi pozwalających na formalną, automatyczną analizę rozważanego procesu. Większość tych rozwiązań jest specyficzna dla dziedziny zastosowań, choć istnieje kilka wyjątków od tej reguły. Należy tu wspomnieć *Sensor Modeling Language*, w skrócie SensorML, oferujący bogatą ontologię sensorów (por. <http://www.sensorml.org> z bogatą dokumentacją). Dualnym przedsięwzięciem jest *Unified Robot Modeling Language*, (URML), z Uniwersytetu w Karlsruhe. Niestety, nie przystaje on zbyt dobrze do naszego projektu, ponieważ URML nie pozwala na reprezentację dynamicznych aspektów działania robotów przemysłowych.

Jeden z ważniejszych kroków naprzód w obszarze formalizacji opisu procesów produkcyjnych uczyniono w NIST, USA, gdzie powstał Process Specification Language [7]. Zarówno ten język, jak i niektóre związane z nim narzędzia, stanowią punkt odniesienia dla prac prowadzonych nad ontologią w projekcie SIARAS.

6. PODSUMOWANIE

W artykule przedstawiono prace nad narzędziami reprezentacji wiedzy (formalizmami i związanymi z nimi algorytmami rozumującymi) przeznaczonymi do użycia w tworzeniu i wykorzystaniu ontologii do adaptacyjnego planowania produkcji w projekcie SIARAS. W pełni automatyczne adaptacyjne planowanie produkcji jest poza zasięgiem obecnych możliwości obliczeniowych, dlatego też zostało przyjęte rozwiązanie wielopoziomowe łączące automatyczne rozumowania, niesymboliczne obliczenia zależne od dziedziny zastosowań i konsultacje z użytkownikiem.

Reprezentacja wiedzy w projekcie SIARAS jest zbudowana wokół pojęcia *ontologii* która to, razem ze specjalizowanymi modułami rozumującymi dołączanymi na żądanie, tworzy jądro *Serwera Umiejętności*. System jest obecnie w późnej fazie projektu; wstępny prototyp już istnieje, a wersja demonstracyjna oczekiwana jest pod koniec 2006.

LITERATURA

- [1] R. J. Brachman, H. J. Levesque. *Knowledge Representation and Reasoning*. Morgan Kaufmann, 2004.
- [2] M. Buchheit, F. M. Donini, A. Schaerf. Decidable reasoning in terminological knowledge representation systems. *J. of Artificial Intelligence Research*, 1993, vol. 1 s. 109–138.
- [3] D. Fensel et al. OIL: An ontology infrastructure for the semantic web. *IEEE Intelligent Systems*, 2001, vol. 16, nr 2, s. 38–45.
- [4] M. Ghallab, D. Nau, P. Traverso. *Automated Planning, Theory and Practice*. Morgan Kaufman, 2004.
- [5] V. Haarslev, R. Möller. Racer: A core inference engine for the semantic web. Dostępny pod: <http://www.franz.com/products/racer/>, 2002.
- [6] I. Horrocks. FaCT and iFaCT. In: *Proceedings of the International Workshop on Description Logics (DL'99)*. Red. P. Lambrix et al, 1999, s. 133–135.
- [7] C. Schlenoff et al. The Process Specification Language (PSL): Overview and version 1.0 specification. Raport instytutowy NISTIR 6459, NIST, Gaithersburg, MD, 2000.
- [8] *The Description Logic Handbook*. Red. F. Baader et al. Cambridge University Press, 2003.
- [9] M. Musen et. al. The Protégé ontology editor and knowledge acquisition system. <http://protege.stanford.edu>, 2006.
- [10] W3C. Semantic web. <http://www.w3.org/2001/sw/>, 2001.
- [11] W3C. Web ontology language (OWL). <http://www.w3.org/2004/OWL/>, 2003.

KNOWLEDGE REPRESENTATION FOR RECONFIGURABLE AUTOMATION SYSTEMS

This article describes the work in progress on knowledge representation formalisms chosen for use in the European project SIARAS. *Skill-Based Inspection and Assembly for Reconfigurable Automation Systems* has a goal of creating intelligent support system for reconfiguration and adaptation of assembly systems. Knowledge is represented in an ontology expressed in OWL, for generic reasoning in Description Logic, and in a number of special-purpose reasoning modules, specific for the application domain.