

WNIOSEK

o finansowanie projektu badawczego

własnego	habilitacyjnego	promotorskiego	1)
----------	-----------------	----------------	----

A. DANE WNIOSKODAWCY

<p>1. Kierownik projektu (imię, nazwisko, tytuł lub stopień naukowy, adres do korespondencji, tel. e-mail) Antoni Ligęza, Prof. dr hab. inż., ul. Zachodnia 5/58, 30-350 Kraków, tel. 0 12-267-44-67, ligeza@agh.edu.pl</p> <p>2. Nazwa i adres jednostki naukowej, telefon, fax, e-mail www */ Akademia Górniczo-Hutnicza Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki, Katedra Automatyki al. Mickiewicza 30, 30-059 Kraków Tel. 0 12-617-28-30, Fax: 0 12-634-15-68 http://www.agh.edu.pl</p> <p>3. Kierownik jednostki Prof. dr hab. Kazimierz Jeleń – Prorektor ds. Nauki</p> <p>4. NIP: 6750001923, REGON: 000001577,</p> <p>5. Nazwa banku, nr rachunku Bank BPH S.A. o/Kraków, ul. Pijarska 1 78 1060 0076 0000 3200 0046 8005</p>	<p>Wypełnia Ministerstwo Nauki i Informatyzacji</p> <p>Nr rejestracyjny wniosku</p> <p>Data wpłynięcia wniosku</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------

*/ Szkoły wyższe podają informacje określone w pkt. 1 w odniesieniu do całej szkoły oraz jednostki podstawowej (zgodnie ze statutem szkoły) planowanej jako miejsce realizacji projektu

A. INFORMACJE OGÓLNE

Tytuł projektu (tytuł powinien w pełni charakteryzować zawartość wniosku – maks. liczba znaków ze spacjami – 250)		Hybrydowa Inżynieria Wiedzy: Zintegrowane narzędzia komputerowe do efektywnego wytwarzania niezawodnego oprogramowania w oparciu o deklaratywny model logiczny. (HEKATE)	
Dyscyplina naukowa (zgodnie z wykazem dziedzin i dyscyplin)		N516: Technologie Informacyjne	
Planowany okres realizacji projektu (w miesiącach)	24	Liczba wykonawców projektu	12
Słowa kluczowe: inżynieria wiedzy, inżynieria oprogramowania, narzędzia komputerowego wspomaganie, logika biznesowa, inteligentne sterowanie			
Planowane nakłady w zł :	Ogółem	Pierwszy rok realizacji projektu	
	340 500	191 000	

Streszczenie projektu: Hybrydowa Inżynieria Wiedzy: Zintegrowane narzędzia komputerowe do efektywnego wytwarzania niezawodnego oprogramowania w oparciu o deklaracyjny model logiczny. (HEKATE)

Akademia Górniczo-Hutnicza Wydz. Elektrotechniki, Automatyki, Informatyki i Elektroniki, Kat. Automatyki

Zapotrzebowanie na złożone, niezawodne i tworzone w ograniczonym czasie oprogramowania nieustannie wzrasta. Pomimo użycia wyrafinowanych technik projektowania nie do uniknięcia pozostaje problem, którym jest jakościowe przejście od abstrakcyjnych metod modelowania struktury programu do poziomu szczegółowości języka programowania. Stosowane powszechnie języki wysokiego poziomu, opierają się niejawnie na koncepcji sekwencyjnego przetwarzania informacji, która jest odległa od pojęciowego i deklaracyjnego modelu programu.

Celem naukowym projektu jest opracowanie narzędzi komputerowego wspomaganie tworzenia oprogramowania, opartych o metody Inżynierii Wiedzy, pozwalających na efektywne budowanie komponentu wnioskującego (tzw. logiki biznesowej). Narzędzia te umożliwią: deklaracyjną reprezentację wiedzy na poziomie specyfikacji projektu, możliwość łatwej modyfikacji kodu na poziomie deklaracyjnej reprezentacji wiedzy, oraz możliwość analizy, ewaluacji i weryfikacji tej wiedzy już w trakcie projektowania, a także intuicyjną, graficzną, hierarchiczną reprezentację wiedzy wspomagającą proces projektowania.

Obszarami zainteresowania wspomnianych narzędzi rozpatrywanymi w niniejszym projekcie są: aplikacje biznesowe, z logiką biznesową na wysokim poziomie abstrakcji, oraz aplikacje inteligentnego sterowania. *Innowacyjność* wskazanych narzędzi będzie polegać na: przyspieszeniu procesu wytwarzania oprogramowania, zmniejszeniu ilości błędów, formalnej weryfikacji w fazie projektowania, łatwości modyfikacji z zachowaniem funkcjonalności oraz poziomu jakości.

Koncepcja proponowanych narzędzi opiera się na rozwinięciu opracowanych wcześniej metod wspomaganie projektowania systemów regulowych, a przede wszystkim: Tabular Trees, Extended Tabular Trees (XTT), oraz Attribute Relationship Diagram (ARD). Podejście to, i stojąca za nim metodologia projektowania i implementacji, będzie nazywane dalej *Hybrydową Inżynierią Wiedzy* (ang. *Hybrid Knowledge Engineering*), oznaczoną akronimem *HEKATE*.

Spodziewane *korzyści* jakie przyniesie projekt to między innymi: skrócenie cyklu rozwijania oprogramowania, dzięki istotnej integracji fazy projektowania i implementacji, jak również zmniejszenie liczby usterek w oprogramowaniu, dzięki użyciu mechanizmów programowania spójnych z metodami projektowania. Zwiększona zostanie niezawodność oprogramowania, dzięki dostarczeniu metod analizy poprawności oprogramowania w fazie projektowania, oraz łatwość modyfikacji projektowanego oprogramowania, dzięki deklaracyjnemu charakterowi metodologii. Zachowana będzie przejrzystość procesu wytwarzania oprogramowania.

Należy zwrócić uwagę, iż dzięki specyfikacji oprogramowania opartej o metody inżynierii wiedzy, oprogramowanie staje się niejako tożsame ze specyfikacją. W proponowanym podejściu wysokopoziomowa specyfikacja oprogramowania staje się "specyfikacją wykonywalną" – kodem, którego analiza, testowanie, uruchomienie i modyfikacja przebiega on-line, pod kontrolą użytkownika końcowego.

Wyniki projektu będą mieć istotne znaczenia dla rozwoju nowych metod w inżynierii oprogramowania i praktycznych zastosowań inżynierii wiedzy. Proponuje się nakreślenie nowych standardów wytwarzania wysoce niezawodnego oprogramowania. Opierać się one będą na deklaracyjnych i wizualnych zarazem metodach projektowania, które z kolei będą posiadały odpowiadający im, dobrze zdefiniowany model logiczny. Koncepcja Hybrydowej Inżynierii Wiedzy opiera się między innymi na rozszerzeniu podejścia MVC (Model-View-Controller), które wprowadziło deklaracyjną klasyfikację elementów funkcjonalnych tworzonego systemu. Głównym atutem MVC jest separacja warstwy logiki biznesowej od prezentacji. W ostatniej dekadzie znalazło ono szerokie zastosowanie przy tworzeniu tzw. aplikacji biznesowych na platformach Java Enterprise Edition. W Hybrydowej Inżynierii Wiedzy proponuje się, w podobny do MVC tj. deklaracyjny sposób opisu struktury systemu oraz deklaracyjny opis logiki biznesowej.

Proponowana integracja metod inżynierii wiedzy i oprogramowania, doprowadzi do wypracowania metodologii bardziej efektywnych, oraz technologii i narzędzi przewyższających obecnie stosowane. Wykorzystanie języka bardzo wysokiego poziomu (Prolog) i integracja występujących w nim mechanizmów ze współcześnie używanymi zaawansowanymi platformami takim jak Java pozwoli na zaproponowanie skuteczniejszych metod budowania oprogramowania biznesowego i systemów inteligentnego sterowania.

Podsumowując, należy uznać, iż rezultaty naukowe i inżynierskie projektu Hekate, pozwolą na zweryfikowanie istniejącego stanu wiedzy co do projektowania i implementacji oprogramowania dla szerokiej klasy systemów informatycznych.

B. INFORMACJE O WYKONAWCACH

Ankieta dorobku naukowego kierownika projektu i najważniejszych wykonawców projektu (na odrębnych stronach dla każdego z wykonawców)

- 1) Imię i nazwisko: **Antoni Ligęza**
- 2) Adres zamieszkania, adres do korespondencji, fax, e-mail, PESEL– wyłącznie dla kierownika projektu: **ul. Zachodnia 5/58, 30-350 Kraków; Katedra Automatyki AGH, al. Mickiewicza 30, 30-059 Kraków; 012-634-15-68; lizeza@agh.edu.pl, PESEL: 56050207256**
- 3) Miejsca zatrudnienia i zajmowane stanowiska: **Akademia Górniczo-Hutnicza, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki, Katedra Automatyki; profesor nadzwyczajny**
- 4) Charakter udziału w realizacji projektu (kierownik lub wykonawca): **kierownik**
- 5) Przebieg pracy naukowej: nazwa szkoły wyższej, instytutu lub innej jednostki organizacyjnej, specjalność, data uzyskania tytułu zawodowego, stopnia naukowego lub tytułu naukowego:

magistra: **Akademia Górniczo-Hutnicza, Wydz. Elektrotechniki, Automatyki, Informatyki i Elektroniki, elektronika s. automatyka, 1980-03-08.**

doktora: **Akademia Górniczo-Hutnicza, Wydz. Elektrotechniki, Automatyki, Informatyki i Elektroniki, doktor nauk technicznych, 1983-06-30.**

doktora habilitowanego: **Akademia Górniczo-Hutnicza, Wydz. Elektrotechniki, Automatyki, Informatyki i Elektroniki, informatyka (sztuczna inteligencja), 1994-05-19.**

profesora: **Akademia Górniczo-Hutnicza, Wydz. Elektrotechniki, Automatyki, Informatyki i Elektroniki, profesor nauk technicznych, 17.05.2006**

- 6) Informacje o pracach wykonanych w okresie ostatnich 4 lat ²⁾ przed zgłoszeniem wniosku (maksimum 10 pozycji, podać miejsce i datę publikacji; opracowane nowe technologie; najważniejsze osiągnięcia poznawcze i zastosowania praktyczne)
 1. Ligęza A.: Logical Foundations for Rule-Based Systems. Książka wydana przez Springer-Verlag, Berlin, Heidelberg, 2006, w serii *Studies in Computational Intelligence 11*. ISBN 3-540-29117-2.
 2. Ligęza A. and P. Fuster Parra: A granular attribute logic for rule-based systems management within tabular trees. *Cybernetic and Systems*, R. Trappl (ed.), Vol. 2, 761-766, Vienna, 2006.
 3. Szwed P. and A. Ligęza: Application of OBDD diagrams in verification of rule-based systems. *Schedae Informaticae*, vol. 14, 2005, 75-94.
 4. Ligęza A. and G.J. Nalepa: Visual design and on-line verification of tabular rule-based systems with XTT. *Lecture Notes in Informatics (LNI)*, Lipsk, 2005, 303-312; Proceedings of LIT-2005, (P. Jantke et al. Eds.), Lipsk, 2005.
 5. Bouzid M. and A. Ligęza: Algebraic temporal specifications with extended TUS : hierarchical granular terms and their applications. In: *ICTAI 2005 : Seventeenth International Conference on Tools with Artificial Intelligence : 14–16 November 2005, Hong Kong, China : Proceedings / ed. Andrew Lim ; IEEE Computer Society, 2005, 249-253.*
 6. Tadeusiewicz R., A.Ligęza and M. Szymkat (Eds.): CMS'05. Vol. 1, Plenary lectures and special sessions : Computer Methods and Systems : V konferencja : 14–16 November 2005, ONT, Kraków, Poland, 452 pp.
 7. Ligęza A. and M. Szpyrka: Reduction of tabular systems. In: *Artificial Intelligence and Soft Computing – ICAISC'2004*, L. Rutkowski et al. (Eds.), *Lecture Notes in Computer Science, LNAI 3070*, 2004, 903-908.

8. Ligęza A.: Selected methods of knowledge engineering in systems diagnosis. A chapter in: *Fault diagnosis : models, artificial intelligence, applications*, J. Korbicz et al. (Eds.), 633-674, Springer,-Verlag, 2004.
9. Ligęza A.: Dual resolution for logical reduction of granular tables. In: *Intelligent Information Systems and Web Mining, Zakopane, 2003*, M.A. Kłopotek et al. (Eds.), *Advances in Soft Computing*, 4240, Springer-Verlag, 2003, 363-372.
10. Ligęza A.: Granular sets and granular relations for algebraic knowledge management . *Proceedings of the Artificial Neural Networks in Engineering conference (ANNiE 2003)*, St. Louis, Missouri, 2003; ASME Press, 169-174.

Pełniejszy wykaz publikacji z ostatnich lat: <http://www.bpp.agh.edu.pl>

Osiągnięcia poznawcze:

- Opracowanie nowych koncepcji zbiorów granularnych i relacji granularnych (poz. 10 wykazu publikacji);
- Sformułowanie koncepcji atrybutowych logik granularnych (pozycja 2 wykazu literatury);
- Sformułowanie szeregu elementów z zakresu reprezentacji wiedzy i wnioskowania oraz sterowanie wnioskowanie w systemach regułowych (poz. 1 wykazu literatury).

Zastosowania praktyczne:

- Projekt i implementacja modułu wnioskowania w języku Prolog (poz. 1 wykazu literatury);
- Koncepcja, współprojektowanie i opracowanie systemów wspomagających zarządzanie pracami dyplomowymi SOPD (sopd.ia.agh.edu.pl) i DYPLOMANT (dypломant.ia.agh.edu.pl), użytkowanymi w KA AGH;
- Konsultowanie i nadzór nad kilkudziesięcioma projektami systemów bazodanowych i internetowych realizowanych głównie w ramach prac dyplomowych.

- 7) Wykonane w okresie ostatnich 4 lat ²⁾ przed zgłoszeniem wniosku i aktualnie realizowane projekty badawcze i celowe finansowane ze środków finansowych na naukę - numery projektów, miejsce realizacji oraz charakter udziału przy realizacji projektu
 1. Uogólnione podejście do zintegrowanego projektowania i implementacji systemów regułowych. Grant KBN nr 4 T11C 027 24 Projekt Mirella, Katedra Automatyki AGH, Kraków, (<http://mirella.ia.agh.edu.pl>) (2003-2004), (kierownik grantu). **Projekt zakończono.**
 2. Zastosowanie metod formalnych do wspomaganie wytwarzania poprawnego oprogramowania systemów czasu rzeczywistego. Grant KBN nr 4 T11C 035 24, Projekt ADDER, Katedra Automatyki AGH, Kraków, (<http://adder.ia.agh.edu.pl>) 2004-2006, (główny wykonawca). **Projekt zakończono.**
 3. Algorytmy diagnostyki procesów przemysłowych uwzględniające problemy uszkodzeń wielokrotnych i dynamiki powstawania symptomów – ALDIAG. Grant MNiI nr, 1527/T11/2005/29, 1.12.2005-30.11.2007, Politechnika Warszawska, Wydział Mechatroniki, Instytut Automatyki i Robotyki, (główny wykonawca).
- 8) Doświadczenia naukowe zdobyte w kraju i za granicą (kraj, instytucja, rodzaj pobytu, okres pobytu, jednostka delegująca)
 - Hiszpania, University of Balearic Islands, visiting professor, 2005-09-01 – 2005-09-30 (1 mies),

- Francja, University of Caen (Normandia), visiting professor, 2005-06-01 – 2005-06-30 (1 mies.),
 - Francja, University of Caen (Normandia), visiting professor, 2004-06-01 – 2004-06-30 (1 mies.),
 - Hiszpania, University of Girona, visiting professor, 1998-11-23 - 1998-12-23 (1 mies.),
 - Hiszpania, University of Girona, visiting professor, 1997-07-01 - 1997-12-23 (6 mies.),
 - Francja, Laboratoire d'Automatique et d'Architecture des Systèmes (LAAS du CNRS), visiting professor, 1996-03-01 - 1996-08-31 (6 mies.),
 - Hiszpania, University of Balearic Islands, visiting professor, 1994-11-01 - 1995-11-30 (1 mies) oraz 1995-06-01 - 1995-07-31 (2 mies.)
 - Francja, Université de Nancy - Centre de Recherche en Informatique de Nancy (CRIN, INRIA), indywidualny staż w ramach Programu TEMPUS, 1994-02-01 - 1994-07-31 (6 mies.)
 - Francja, Laboratoire d'Automatique et d'Architecture des Systèmes (LAAS du CNRS), staż w ramach Programu TEMPUS, 1992-02-01 - 1992-07-31 (6 mies),
 - Dania, Technical University of Lyngby, staż naukowy, 1988-09-01 - 1988-12-23 (4 mies.)
- 9) Najważniejsze międzynarodowe i krajowe wyróżnienia wynikające z prowadzenia badań naukowych lub prac rozwojowych (rodzaj wyróżnienia, miejsce i data)
- Indywidualna Nagroda Ministra za książkę “*Logical Foundations for Rule-Based Systems*”, wydanie z 2005 r.
 - Powołanie na członka stałej Komisji Nauk Technicznych PAU, 2004.
 - Zespołowa Nagroda Ministra Edukacji Narodowej i Sportu (współautorstwo monografii “*Diagnostyka Procesów*”, pod red. J. Korbicza i in.), 2003.
 - Nagrody Naukowe Rektora AGH: 1986, 1988, 1989, 1990, 1993, 1994, 1995, 1996, 1997, 1999, 2000, 2001, 2002, 2003, 2004.
 - Członkostwo IEEE Computer Society i ACM; zaproszenie do zmiany statusu członka IEEE na Senior Member.

Oświadczenie

Przyjmuję warunki udziału w konkursie projektów badawczych, określone w przepisach w sprawie kryteriów i trybu przyznawania i rozliczania środków finansowych na naukę oraz wyrażam zgodę na zamieszczenie moich danych osobowych zawartych we wniosku w zbiorze danych Ministerstwa Nauki i Informatyzacji oraz na przetwarzanie tych danych zgodnie z przepisami ustawy z dnia 29 sierpnia 1997 r. o ochronie danych osobowych (Dz. U. Nr 133, poz. 883 z późn. zm.)

Miejscowość i data

Podpis osoby, której dotyczy ankieta

Ankieta dorobku naukowego kierownika projektu i najważniejszych wykonawców projektu (na odrębnych stronach dla każdego z wykonawców)

- 1) Imię i nazwisko: **Grzegorz J. Nalepa**
- 2) Adres zamieszkania, adres do korespondencji, fax, e-mail, PESEL– wyłącznie dla kierownika projektu: **ul. Kobierzyńska 98/40, 30-382 Kraków, Al. Mickiewicza 30, 30-059 Kraków, gjn@agh.edu.pl**
- 3) Miejsca zatrudnienia i zajmowane stanowiska: **Akademia Górniczo-Hutnicza, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki, Katedra Automatyki; adiunkt.**
- 4) Charakter udziału w realizacji projektu (kierownik lub wykonawca): **wykonawca**
- 5) Przebieg pracy naukowej: nazwa szkoły wyższej, instytutu lub innej jednostki organizacyjnej, specjalność, data uzyskania tytułu zawodowego, stopnia naukowego lub tytułu naukowego:
magistra: **Akademia Górniczo-Hutnicza, Wydz. Elektrotechniki Automatyki, Informatyki i Elektroniki, kierunek: automatyka i robotyka, spec. informatyka w sterowaniu i zarządzaniu, 21.6.1999**
doktora: **Akademia Górniczo-Hutnicza, Wydz. Elektrotechniki Automatyki, Informatyki i Elektroniki, 28.10.2004, dyscyplina Informatyka, specjalność Sztuczna Inteligencja, praca obroniona z wyróżnieniem, Rozprawa: „Meta-Level Approach to Integrated Process of Design and Implementation of Rule-Based Systems”.**
- 6) Informacje o pracach wykonanych w okresie ostatnich 4 lat ²⁾ przed zgłoszeniem wniosku (maksimum 10 pozycji, podać miejsce i datę publikacji; opracowane nowe technologie; najważniejsze osiągnięcia poznawcze i zastosowania praktyczne)

Publikacje:

1. Nalepa G. J., Ligęza A.: Prolog-based analysis of tabular rule-based systems with the XTT approach, W: FLAIRS 2006 : proceedings of the nineteenth international Florida Artificial Intelligence Research Society conference, / eds. Geoffrey C. J. Sutcliffe, Randy G. Goebel ; FLAIRS, Menlo Park : AAAI Press, 2006. S. 426-431.
2. Nalepa G. J., Zięcik P.: Integrated embedded prolog platform for rule-based control systems, W: MIXDES 2006 : MIXed DESign of integrated circuits and systems : proceedings of the international conference : Gdynia, Poland 22-24 June 2006 / ed. Andrzej Napieralski., Technical University Lodz. Department of Microelectronics and Computer Science, 2006, S. 716-721.
3. Nalepa G. J., Ligęza A.: Conceptual modelling and automated implementation of rule-based systems, W: Software engineering : evolution and emerging technologies / eds. Krzysztof Zieliński, Tomasz Szmuc. Amsterdam : IOS Press, 2005. (Frontiers in Artificial Intelligence and Applications ; vol. 130). S. 330-340.
4. Nalepa G. J., Ligęza A.: Rule-based systems design and implementation : methodologies and technologies, W: CMS'05. Vol. 1, Plenary lectures and special session papers : Computer Methods and Systems : V konferencja : 14-16 November 2005, Kraków, Poland / eds. Ryszard Tadeusiewicz, Antoni Ligęza, Maciej Szymkat ; AGH University of Science and Technology Cracow, Jagiellonian University, Cracow University of Technology. Kraków : Oprogramowanie Naukowo-Techniczne, 2005. S. 329-340.
5. Nalepa G. J., Ligęza A.: Security systems design and analysis using an integrated rule-based systems approach, W: Advances in Web Intelligence : third international Atlantic Web Intelligence Conference AWIC 2005 : Lodz, Poland, June 6-9, 2005 : proceedings / eds. Piotr S. Szczepaniak, Janusz Kacprzyk, Adam Niewiadomski. Berlin ; Heidelberg : Springer-Verlag, 2005. (Lecture Notes in Artificial Intelligence ; 3528. Subseries of Lecture Notes in Computer Science). S. 334-340.
6. Nalepa G. J., Ligęza A.: A visual edition tool for design and verification of knowledge in rule-based systems, Systems Science, 2005 vol. 31 no. 3 s. 103-109.

7. Nalepa G. J., Ligęza A.: A graphical tabular model for rule-based logic programming and verification, *Systems Science*, 2005 vol. 31 no. 2 s. 89-95.
8. Nalepa G. J., Szpyrka M., Two formal approaches to design and verification of embedded rule-based systems, W: *Real-Time Programming 2004 (WRTP 2004) : a proceedings volume from the 28th IFAC/IFIP Workshop on Real-Time Programming, WRTP 2004 and the International Workshop on Software Engineering, IWSS 2004 : Istanbul, Turkey, 8-10 September 2004 / eds. M. Colnaric, W. A. Halang, M. Węgrzyn. Oxford : Elsevier Ltd, 2005. (Elsevier IFAC Publications). S. 91-96.*
9. Nalepa G. J., Ligęza A.: Designing reliable web security systems using rule-based systems approach, W: *Advances in Web Intelligence : first international Atlantic Web Intelligence Conference AWIC 2003 : Madrid, Spain, May 5-6, 2003 : proceedings / eds. Ernestina Menasalvas, Javier Segovia, Piotr S. Szczepaniak. Berlin ; Heidelberg : Springer-Verlag, 2003. (Lecture Notes in Computer Science) ; (Lecture Notes in Artificial Intelligence ; 2663). S. 124-133.*
10. Nalepa G. J., Ligęza A.: Graphical CASE tools for integrated design and verification of rule-based systems, W: *AI-METH 2002 : METHods of Artificial Intelligence : 13-15 November, Gliwice, Poland : proceedings / eds. T. Burczyński, W. Cholewa, W. Moczulski ; Silesian University of Technology. Department for Strength of Materials and Computational Mechanics. Department of Fundamentals of Machinery Design, Polish Association for Computational Mechanics. Gliwice : SUT DSMCM DFMD, 2002. S. 307-313.*

Pełniejszy wykaz publikacji z ostatnich lat: <http://www.bpp.agh.edu.pl>

Osiągnięcia poznawcze:

- Koncepcja EPP (Embedded Prolog Platform), zintegrowanego środowiska uruchomieniowego języka Prolog dla systemów wbudowanych (publikacja 2).
- Koncepcja XTT (EXtended Tabular Trees) nowej tablicowo-tabelarycznej metody reprezentacji wiedzy dla regułowych systemów ekspertowych. (publikacje 4, 7 i 10).
- Koncepcja użycia deklaratywnych metod projektowania i formalnej analizy systemów bezpieczeństwa (publikacje 5 i 9).

Zastosowania praktyczne:

- Prototyp EPP na platformie Palm Pilot (publikacja 2)
- Prototyp narzędzia CASE wspierającego projektowanie regułowych systemów ekspertowych (publikacja 6, 8, 10), ich prototypowanie w języku Prolog (publikacja 3, 4), oraz formalną analizę (publikacja 1).

- 7) Wykonane w okresie ostatnich 4 lat ²⁾ przed zgłoszeniem wniosku i aktualnie realizowane projekty badawcze i celowe finansowane ze środków finansowych na naukę - numery projektów, miejsce realizacji oraz charakter udziału przy realizacji projektu
 1. Uogólnione podejście do zintegrowanego projektowania i implementacji systemów regułowych. Grant KBN nr 4 T11C 027 24 Projekt Mirella, Katedra Automatyki AGH, Kraków, (<http://mirella.ia.agh.edu.pl>) (2003-2004), (główny wykonawca). **Projekt zakończono.**
 2. Zastosowanie metod formalnych do wspomaganie wytwarzania poprawnego oprogramowania systemów czasu rzeczywistego. Grant KBN nr 4 T11C 035 24, Projekt ADDER, Katedra Automatyki AGH, Kraków, (<http://adder.ia.agh.edu.pl>) 2004-2006, (wykonawca). **Projekt zakończono.**
- 8) Doświadczenia naukowe zdobyte w kraju i za granicą (kraj, instytucja, rodzaj pobytu, okres pobytu, jednostka delegująca)
- 9) Najważniejsze międzynarodowe i krajowe wyróżnienia wynikające z prowadzenia badań naukowych lub prac rozwojowych (rodzaj wyróżnienia, miejsce i data)

Druga nagroda w XVI Ogólnopolskim Konkursie na najlepsze prace magisterskie z informatyki w roku 1999, organizowanym przez Polskie Towarzystwo Informatyczne, za

pracę pt. *Graphical user interface to Kheops rule-based expert system, (Interfejs graficzny wspomagający użytkowanie ekspertowego systemu regułowego Kheops)*, Wrocław, 22.12.1999

Oświadczenie

Przyjmuję warunki udziału w konkursie projektów badawczych, określone w przepisach w sprawie kryteriów i trybu przyznawania i rozliczania środków finansowych na naukę oraz wyrażam zgodę na zamieszczenie moich danych osobowych zawartych we wniosku w zbiorze danych Ministerstwa Nauki i Informatyzacji oraz na przetwarzanie tych danych zgodnie z przepisami ustawy z dnia 29 sierpnia 1997 r. o ochronie danych osobowych (Dz. U. Nr 133, poz. 883 z późn. zm.)

Miejscowość i data

Podpis osoby, której dotyczy ankieta

Ankieta dorobku naukowego kierownika projektu i najważniejszych wykonawców projektu (na odrębnych stronach dla każdego z wykonawców)

- 1) Imię i nazwisko: **Igor Wojnicki**
- 2) Adres zamieszkania, adres do korespondencji, fax, e-mail, PESEL – wyłącznie dla kierownika projektu: **ul. Pszona 22/18, 31-462 Kraków; Katedra Automatyki AGH, al. Mickiewicza 30, 30-059 Kraków; wojnicki@agh.edu.pl; tel. 508 444450**
- 3) Miejsca zatrudnienia i zajmowane stanowiska: **Akademia Górniczo-Hutnicza, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki, Katedra Automatyki; adiunkt**
- 4) Charakter udziału w realizacji projektu (kierownik lub wykonawca): **wykonawca**
- 5) Przebieg pracy naukowej: nazwa szkoły wyższej, instytutu lub innej jednostki organizacyjnej, specjalność, data uzyskania tytułu zawodowego, stopnia naukowego lub tytułu naukowego:

magistra: **Akademia Górniczo-Hutnicza, Wydz. Elektrotechniki, Automatyki, Informatyki i Elektroniki, Automatyka i Robotyka s. Informatyka w Sterowaniu i Zarządzaniu, 31.05.2000.**

doktora: **Akademia Górniczo-Hutnicza, Wydz. Elektrotechniki, Automatyki, Informatyki i Elektroniki, Doktor Nauk Technicznych w Dyscyplinie Informatyka, 28.10.2004,**

Rozprawa „A Rule-based Inference Engine Extending Knowledge Processing Capabilities of Relational Database Management Systems”.

- 6) Informacje o pracach wykonanych w okresie ostatnich 4 lat ²⁾ przed zgłoszeniem wniosku (maksimum 10 pozycji, podać miejsce i datę publikacji; opracowane nowe technologie; najważniejsze osiągnięcia poznawcze i zastosowania praktyczne)
 1. Jelly View – przetwarzanie wiedzy intensjonalnej w relacyjnych bazach danych, Igor WOJNICKI: Inżynieria wiedzy i systemy ekspertowe. T. 1 / pod red. Adama Grzecha. — Wrocław : Oficyna Wydawnicza Politechniki Wrocławskiej, 2006. VI Krajowa Konferencja Naukowa „Inżynieria Wiedzy i Systemy Ekspertowe” (Wrocław, 21–23 czerwca 2006 roku). — S. 237–243. — Bibliogr. s. 242–243
 2. Intensional Knowledge Processing in RDBMS - Improved Performance of the ReDaReS System, Igor WOJNICKI, International MultiConference of Engineers and Computer Scientists 2006, Hong Kong, 20-22 June 2006, International Workshop on Artificial Intelligence and Applications, IAENG, ISBN: 988-98671-3-3
 3. Jelly View - A Technology for Arbitrarily Advanced Queries within RDBMS, Igor WOJNICKI, Antoni LIGEZA, SAC 2005, The 20th Annual ACM Symposium on Applied Computing Santa Fe, New Mexico, 13-17 Marzec 2005.
 4. Handling Recursive Queries within RDBMS with Jelly View Technology. Some Experimental Results with the ReDaReS System, Igor WOJNICKI, Antoni LIGEZA, Konferencja KAM 2005, Złotniki, Poland, w Pozyskiwanie wiedzy i zarządzanie wiedzą, red: Małgorzata Nycz, Mieczysław Lech Owoc, Wrocław 2005.
 5. Jelly View: Extending Relational Database Management Systems toward Deductive Databases, Igor WOJNICKI, Computer Science AGH Journal 2004.
 6. Extending Data Processing Capabilities of Relational Database Management Systems. Igor WOJNICKI, Cezary Z. JANIKOW. International Conference on Artificial Intelligence, Las Vegas, Nevada, USA, 2003.
 7. Relational Database Rule System. Igor WOJNICKI, Antoni LIGEZA. The 6th World Multiconference on Systemics, Cybernetics and Informatics. Orlando, FL, USA, 14-18 Lipiec 2002.

Osiągnięcia poznawcze:

- Opracowanie koncepcji *Jelly View* rozszerzającej możliwości przetwarzania informacji w Relacyjnych Bazach Danych.

Zastosowania praktyczne:

- Implementacja prototypowego systemu ReDaReS realizującego koncepcję *Jelly View*.
 - Koordynacja projektowania oraz wdrożenie systemu rezerwacji i wypożyczenia urządzeń przenośnych w KA.
- 7) Wykonane w okresie ostatnich 4 lat ²⁾ przed zgłoszeniem wniosku i aktualnie realizowane projekty badawcze i celowe finansowane ze środków finansowych na naukę - numery projektów, miejsce realizacji oraz charakter udziału przy realizacji projektu
- 8) Doświadczenia naukowe zdobyte w kraju i za granicą (kraj, instytucja, rodzaj pobytu, okres pobytu, jednostka delegująca)
- USA, University of St.Louis - Missouri, Visiting Assistant Professor, 2004/08/12-2005/08/12
 - USA, Southern Illinois University Edwardsville, Lecturer, 2003/08/15-2004/08/11
 - USA, University of St.Louis - Missouri, Lecturer (Visiting Professor), 2002/08/15-2003/08/15
- 9) Najważniejsze międzynarodowe i krajowe wyróżnienia wynikające z prowadzenia badań naukowych lub prac rozwojowych (rodzaj wyróżnienia, miejsce i data)
- Członkostwo w International Association of Engineers (IAENG).

Oświadczenie

Przyjmuję warunki udziału w konkursie projektów badawczych, określone w przepisach w sprawie kryteriów i trybu przyznawania i rozliczania środków finansowych na naukę oraz wyrażam zgodę na zamieszczenie moich danych osobowych zawartych we wniosku w zbiorze danych Ministerstwa Nauki i Informatyzacji oraz na przetwarzanie tych danych zgodnie z przepisami ustawy z dnia 29 sierpnia 1997 r. o ochronie danych osobowych (Dz. U. Nr 133, poz. 883 z późn. zm.)

Miejscowość i data

Podpis osoby, której dotyczy ankieta

Ankieta dorobku naukowego kierownika projektu i najważniejszych wykonawców projektu (na odrębnych stronach dla każdego z wykonawców)

- 1) Imię i nazwisko: **Konrad Kułakowski**
- 2) Adres zamieszkania, adres do korespondencji, fax, e-mail, PESEL – wyłącznie dla kierownika projektu: **ul. Radzikowskiego 27/2, 31-315 Kraków; kkułak@agh.edu.pl; tel. 12 638 0506, 607 614 372**
- 3) Miejsca zatrudnienia i zajmowane stanowiska:
 1. **Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki, Katedra Automatyki; adiunkt**
 2. **Sabre Polska; inżynier oprogramowania**
- 4) Charakter udziału w realizacji projektu (kierownik lub wykonawca): **wykonawca**
- 5) Przebieg pracy naukowej: nazwa szkoły wyższej, instytutu lub innej jednostki organizacyjnej, specjalność, data uzyskania tytułu zawodowego, stopnia naukowego lub tytułu naukowego:

Magister Informatyki. **Praca magisterska „Music Grammars” obroniona z wyróżnieniem na Uniwersytecie Jagiellońskim. Data otrzymania tytułu: 11 czerwiec 1999.**

Doktor nauk technicznych, **specjalność informatyka. Rozprawa „Konstrukcja i analiza oprogramowania sterowników wspomaganie metodami formalnymi” obroniona z wyróżnieniem na Akademii Górniczo-Hutniczej im. Stanisława Staszica w Krakowie. Data otrzymania stopnia: 29 kwiecień 2004.**

- 6) Informacje o pracach wykonanych w okresie ostatnich 4 lat ²⁾ przed zgłoszeniem wniosku (maksimum 10 pozycji, podać miejsce i datę publikacji; opracowane nowe technologie; najważniejsze osiągnięcia poznawcze i zastosowania praktyczne)
 1. A practical approach to the modelling, visualising and executing of reactive systems, M. Kostrzewa, K. Kułakowski, MIXDES 2006 - MIXed DESign of integrated circuits and systems : proceedings of the international conference : Gdynia, Poland 22–24 June 2006 ed. Andrzej Napieralski.
 2. Wspomaganie tworzenia oprogramowania systemów reaktywnych w Real-Time Java, K. Kułakowski, M. Kostrzewa, SCR'06, Ustroń, 11-14 września 2006 (praca przyjęta na konferencje).
 3. Real-Time Java – platforma programistyczna dla systemów czasu rzeczywistego, K. Kułakowski, SCR'05, Ustroń, 12-15 września 2005
 4. Weryfikacja zgodności projektu ze specyfikacją z wykorzystaniem sieci Petriego i Algebry Procesów CCS, K. Kułakowski, SCR'02, Ustroń, 16–19 września 2002
- 7) Wykonane w okresie ostatnich 4 lat ²⁾ przed zgłoszeniem wniosku i aktualnie realizowane projekty badawcze i celowe finansowane ze środków finansowych na naukę - numery projektów, miejsce realizacji oraz charakter udziału przy realizacji projektu

Prowadząc badania w ramach grantu KBN 4 T11C 035 24 opracowałem metodę pozwalającą transformować modele oparte o Sieci Petriego do równoważnej im postaci algebraicznej CCS. Wyniki te zostały zawarte w mojej rozprawie doktorskiej.
- 8) Doświadczenia naukowe zdobyte w kraju i za granicą (kraj, instytucja, rodzaj pobytu, okres pobytu, jednostka delegująca)
- 9) Najważniejsze międzynarodowe i krajowe wyróżnienia wynikające z prowadzenia badań naukowych lub prac rozwojowych (rodzaj wyróżnienia, miejsce i data)

Oświadczenie

Przyjmuję warunki udziału w konkursie projektów badawczych, określone w przepisach w sprawie kryteriów i trybu przyznawania i rozliczania środków finansowych na naukę oraz wyrażam zgodę na zamieszczenie moich danych osobowych zawartych we wniosku w zbiorze danych Ministerstwa Nauki i Informatyzacji oraz na przetwarzanie tych danych zgodnie z przepisami ustawy z dnia 29 sierpnia 1997 r. o ochronie danych osobowych (Dz. U. Nr 133, poz. 883 z późn. zm.)

Miejscowość i data

Podpis osoby, której dotyczy ankieta

D. OPIS PROJEKTU BADAWCZEGO, METODYKA BADAŃ ORAZ CHARAKTERYSTYKA OCZEKIWANYCH WYNIKÓW

(maks. do 10 stron standardowego maszynopisu)

1. Cel naukowy projektu (jaki problem naukowy wnioskodawca podejmuje się rozwiązać, co jest jego istotą, dokładna charakterystyka efektu końcowego)

1.1 Jaki problem naukowy wnioskodawca podejmuje się rozwiązać?

Inżynieria współczesnych systemów informatycznych jest dziedziną której złożoność nieustannie wzrasta. Jest to wynikiem dwóch nakładających się tendencji: wzrastających wymagań użytkownika i wzrastającej funkcjonalności oraz mocy obliczeniowej systemów komputerowych. Jest to również dziedzina rozwijająca się na dzień dzisiejszy dwutorowo. Pierwszym wątkiem jest rozwijanie technologii sprzętowych, drugim zaś prace nad inżynierią oprogramowania. Warty odnotowania jest fakt, iż rozwój obu tych obszarów przebiega w dużym stopniu asynchronicznie.

Zapotrzebowanie na złożone, niezawodne i tworzone w ograniczonym czasie oprogramowania nieustannie wzrasta. W tym kontekście podstawowym problemem inżynierii oprogramowania jest dostarczanie odpowiednich *metod i narzędzi* tworzenia takiego oprogramowania. Metody te powinny zapewnić niezależność tworzonego oprogramowania od szybko postępujących zmian platform sprzętowych. Ponadto muszą zapewniać poprawną pracę tworzonego kodu na maksymalnie dużej liczbie platform sprzętowo-programowych. Dlatego też jednym z podstawowych charakterystyk tych narzędzi jest *abstrahowanie od specyfiki sprzętu* na którym implementowane jest projektowane oprogramowanie. Mają temu między innymi służyć metodologie analizy i projektowania oprogramowania takie jak szeroko rozumiane tzw. podejście obiektowe, w tym język i metodologia UML [1]. Oraz technologie wirtualnych środowisk uruchomieniowych takich jak JVM, CLI, LLVM (<http://llvm.org>), HLVM (<http://hlvm.org>). Wspomniane metodologie są często wspierane narzędziami typu CASE np. wiodący na rynku pakiet oprogramowania implementujący metodologię UML: IBM Rational Software.

Ważnym aspektem jest umożliwienie łatwej modyfikacji tworzonego oprogramowania przy zmieniających się wymaganiach funkcjonalnych. Trzeba mieć świadomość, że wyjściowa specyfikacja wymagań *ewoluuje* zazwyczaj *w trakcie trwania projektu*. Zmieniające się wymagania, w tym rosnąca ich ilość jest częstą przyczyną istotnych opóźnień w realizacji systemów informatycznych.

Pomimo użycia wyrafinowanych technik nie do uniknięcia pozostaje problem, którym jest jakościowe przejście od *abstrakcyjnych metod modelowania struktury programu* (często o charakterze deklaratywnym) do poziomu szczegółowości *języka programowania*, którego elementy składniowe zbliżają kod do postaci uruchamialnej przez maszynę Turinga. Stosowane powszechnie języki wysokiego poziomu, w tym tzw. języki obiektowe, opierają się niejawnie na koncepcji sekwencyjnego przetwarzania informacji, która jest odległa od pojęciowego i deklaratywnego modelu programu.

Celem naukowym projektu jest opracowanie prototypowych narzędzi komputerowego wspomaganie tworzenia oprogramowania, opartych o metody Inżynierii Wiedzy (a w szczególności o programowanie deklaratywne), pozwalających na efektywne budowanie komponentu wnioskującego (tzw. logiki biznesowej). Projektowane narzędzia umożliwią:

- deklaratywną reprezentację wiedzy na poziomie specyfikacji projektu,
- uniezależnienie (abstrakcję) od platformy sprzętowej, a więc łatwą przenaszalność,
- możliwość łatwej modyfikacji kodu na poziomie deklaratywnej reprezentacji wiedzy,
- możliwość analizy, ewaluacji i weryfikacji tej wiedzy już w trakcie projektowania,
- intuicyjną, graficzną, hierarchiczną reprezentację wiedzy.

Proponowane narzędzia będą składały się z następujących modułów: wizualnego edytora bazy wiedzy, analizy i weryfikacji, uruchomieniowego, interfejsu z bazą wiedzy, oraz modułów translacyjnych. Funkcjonalność systemu może być łatwo modyfikowana za pomocą dodatkowych modułów. Obszarami zainteresowania wspomnianych narzędzi rozpatrywanymi w niniejszym projekcie są:

- aplikacje biznesowe, z logiką biznesową na wysokim poziomie abstrakcji (z wykorzystaniem specyfikacji Java Enterprise Edition i technologii JBoss, czy też IBM WebSphere, zbioru technologii LAMP, czy też Ruby on Rails),

- aplikacje inteligentnego monitorowania i sterowania, włączając w to systemy autonomiczne (sterowanie robotów mobilnych).

Projektowanie aplikacji w obu tych obszarach jest szczególnie predestynowane do wykorzystania metod deklaratywnych, jako bazowego paradygmatu specyfikacji wiedzy. Innowacyjność wskazanych narzędzi będzie polegać na:

- przyspieszeniu procesu wytwarzania oprogramowania,
- zmniejszeniu ilości błędów,
- formalną weryfikację w fazie projektowania,
- łatwość modyfikacji z zachowaniem funkcjonalności oraz poziomu jakości.

1.2 Co jest istotą problemu?

Istotą wskazanego powyżej problemu jest niejawne oparcie się na dualizmie: sprzęt-oprogramowanie, a co za tym idzie trudność przejścia od conceptualnych metod projektowania oprogramowania (abstrahujących od specyfiki sprzętu), do poziomu języka programowania (jednoznacznie przekładalnego do postaci uruchamialnej na konkretnym sprzęcie typu maszyna Turinga). Tak więc rozwiązywany problem jest konsekwencją braku "gładkiego zszycia" albo "niedopasowania" pomiędzy istniejącymi metodologiami modelowania systemów i projektowania oprogramowania, a implementacją programu na konkretnej architekturze.

Warto tu zwrócić uwagę na alternatywne metody modelowania systemów informatycznych - nazywane kiedyś "podejściem cybernetycznym" - rozwijane obecnie w pewnym zakresie w ramach inżynierii wiedzy [2] i szerzej rozumianej sztucznej inteligencji [3, 4, 5, 6].

Inżynieria wiedzy stanowi praktyczny wymiar sztucznej inteligencji. Jest dziedziną zorientowaną na projektowanie i implementację systemów informatycznych, opartych o teorie i koncepcje szeroko rozumianej sztucznej inteligencji. Dobrym przykładem są tutaj złożone systemy regułowe i wypracowane narzędzia do ich tworzenia (np. Kheops, X-Pert, G-2, Clips, Jess).

Podstawą inżynierii wiedzy są deklaratywne metody reprezentacji wiedzy, pozwalające na uchwycenie wiedzy o konstruowanym systemie w sposób pozwalający na jej efektywne przetwarzanie przez specjalizowane mechanizmy wnioskujące. Mechanizmy te były i są konstruowane przy pomocy języków programowania diametralnie odmiennych od popularnych dziś języków programowania ogólnego zastosowania. Najbardziej znane to opierający się na reprezentacji symbolicznej Lisp, i przede wszystkim mający silne podstawy logiczne deklaratywny Prolog [7].

W inżynierii wiedzy również wykorzystuje się drogę abstrakcji jako jedną z metod tworzenia systemu. W przeciwieństwie jednak do inżynierii oprogramowania nie abstrahuje się tu od specyfiki sprzętu na którym ma pracować oprogramowanie, lecz opisuje rozpatrywany system na różnych poziomach szczegółowości i dokładności [8]. W inżynierii wiedzy traktuje się system holistycznie, w przeciwieństwie do dualistycznego podejścia inżynierii oprogramowania. Co istotne, w przejściu od najbardziej ogólnej reprezentacji wiedzy do poziomu projektu fizycznego, stanowiącego implementację, nie ma, występującego w inżynierii oprogramowania, "przejścia fazowego" pomiędzy projektem conceptualnym, a językiem programowania. Języki i środowiska uruchomieniowe wykorzystywane w inżynierii wiedzy, przede wszystkim język Prolog, opierają się bowiem na jawnej, deklaratywnej reprezentacji wiedzy.

Można więc zaryzykować twierdzenie, iż "luka semantyczna" (ang. *the semantic gap*) występująca pomiędzy metodami projektowania i implementacji, jest w inżynierii wiedzy drastycznie mniejsza, niż to ma miejsce w przypadku współczesnej inżynierii oprogramowania. Istnienie i stałe powiększanie się tej luki semantycznej jest jednym z istotnych czynników warunkujących rosnące trudności z efektywnym projektowaniem poprawnego oprogramowania.

1.3. Dokładna charakterystyka efektu końcowego

Podstawowym efektem końcowym projektu ma być zaproponowanie pakietu prototypowych narzędzi projektowania systemów informatycznych. Narzędzia te będą wykorzystywały wybrane mechanizmy inżynierii wiedzy (w tym metody reprezentacji wiedzy i programowania deklaratywnego) do efektywnego konstruowania kluczowych fragmentów współczesnego oprogramowania we wskazanych obszarach. Będą one wspierały zintegrowany, wielowarstwowy proces projektowania, analizy i implementacji systemów informatycznych [9]. Wykorzystane tu zostaną doświadczenia nabyte w trakcie opracowywania sprawdzonych narzędzi takich jak:

Mirella [10], ReDaReS [11], QaRDE i innych starszych systemów. Proponowany zestaw narzędzi będzie zawierał następujące komponenty:

- wizualny edytor konceptualny, umożliwiający sprecyzowanie struktury systemu, uwzględniając wiele hierarchicznych poziomów szczegółowości,
- wizualny edytor logiczny, za pomocą którego zostaje określona logika działania komponentów zdefiniowanych przez edytor konceptualny,
- moduł analizy i weryfikacji, który umożliwia formalną analizę i weryfikację tworzonego oprogramowania,
- moduł uruchomieniowy, który jest wieloplatformowym środowiskiem uruchamiania aplikacji,
- moduł bazy wiedzy, która reprezentuje aplikację, oraz
- moduły translacyjne, umożliwiające translację bazy wiedzy oraz jej integrację z innymi językami programowania.

Koncepcja proponowanych narzędzi opiera się na rozwinięciu opracowanych wcześniej metod wspomagania projektowania systemów regułowych, a przede wszystkim: Tabular Trees [12] (zostanie wykorzystane przez edytor logiczny), Extended Tabular Trees (XTT) [10] (zostanie również wykorzystane przez edytor logiczny) oraz Attribute Relationship Diagram (ARD) [6,9] (zostanie wykorzystane przez edytor konceptualny).

Wymienione narzędzia zostaną użyte do wytwarzania dwóch podstawowych klas oprogramowania:

- możliwie „odległego” od sprzętu oprogramowanie klasy biznes (operującego na poziomie "logiki problemu"), np. z wykorzystaniem specyfikacji JavaEE, oraz
- bliskiego sprzętowi oprogramowania komputerowych systemów sterowania (operujące na poziomie "logiki sprzętu"), ze szczególnym przeznaczeniem dla inteligentnego sterowania systemów mobilnych, z wykorzystaniem Platformy Mobilnego Robota Hexor.

Podejście to, i stojąca za nim metodologia projektowania i implementacji, będzie nazywane dalej Hybrydową Inżynierią Wiedzy (ang. *Hybrid Knowledge Engineering*), oznaczoną akronimem *HEKATE*.

Spodziewane korzyści jakie przyniesie projekt Hekate to między innymi:

- skrócenie cyklu rozwijania oprogramowania, dzięki istotnej integracji fazy projektowania i implementacji,
- zmniejszenie liczby usterek i błędów w oprogramowaniu, dzięki użyciu mechanizmów programowania spójnych z metodami projektowania, KKIO
- zwiększenie niezawodności oprogramowania, dzięki dostarczeniu metod analizy poprawności oprogramowania w fazie projektowania,
- łatwość modyfikacji i rozszerzania projektowanego oprogramowania, dzięki deklaracywnemu charakterowi metodologii,
- przejrzystość procesu wytwarzania oprogramowania we wszystkich jego fazach,
- potencjalne wsparcie dla zautomatyzowania tworzenia zaawansowanych interfejsów użytkownika, poprzez automatyczne generowanie interfejsu na podstawie specyfikacji.

Należy zwrócić uwagę, iż dzięki specyfikacji oprogramowania opartej o metody inżynierii wiedzy, oprogramowanie staje się niejako tożsame ze specyfikacją. W proponowanym podejściu wysokopoziomowa specyfikacja oprogramowania staje się "specyfikacją wykonywalną" – kodem, którego analiza, testowanie, uruchomienie i modyfikacja przebiega on-line, pod kontrolą użytkownika końcowego.

2. Znaczenie projektu (co uzasadnia podjęcie tego problemu w kraju, jakie przesłanki skłaniają wnioskodawcę do podjęcia proponowanego tematu, dlaczego projekt zdaniem autora powinien być finansowany, znaczenie wyników projektu dla rozwoju danej dziedziny i dyscypliny naukowej oraz rozwoju cywilizacyjnego, czy w przypadku pozytywnych wyników będą one mogły znaleźć praktyczne zastosowanie)

2.1. Co uzasadnia podjęcie tego problemu w kraju?

W kraju mamy do czynienia z licznymi badaniami i doświadczeniami w dziedzinie Inżynierii Wiedzy oraz Inżynierii Oprogramowania (Konferencja KKIO). Należy również zauważyć burzliwy rozwój firm z sektora IT w Polsce, które to firmy stanowią istotną i innowacyjną część gospodarki. Z uwagi na powyższe fakty, wydaje się być uzasadnionym oraz celowym rozwijanie zaproponowanych w niniejszym projekcie koncepcji i technologii.

Podjęcie tej problematyki jest naturalną kontynuacją badań prowadzonych w tych dziedzinach w Katedrze Automatyki AGH, oraz licznych i udokumentowanych doświadczeń z użyciem metod formalnych jakie mają pracownicy Katedry należący do wnioskującego zespołu (zakończone projekty badawcze finansowane przez KBN, w szczególności granty: Regulus (Nr: 8 T11C 01917), Mirella (Nr: 4 T11C 027 24) i Adder (Nr: 4 T11C 035 24)). Badania w w/w dziedzinach są kontynuowane, a ich wyniki były w roku bieżącym prezentowane na konferencjach międzynarodowych za granicą [13] i w kraju [14].

2.2. Jakie przesłanki skłaniają wnioskodawcę do podjęcia proponowanego tematu?

Do podjęcia proponowanego tematu wnioskodawców skłaniają między innymi następujące przesłanki:

- bogate doświadczenia wnioskującego zespołu w dziedzinie inżynierii wiedzy,
- metod formalnych,
- inżynierii oprogramowania,
- wskazane na początku problemy związane z efektywnym projektowaniem i implementacją oprogramowania
- wypracowanie nowatorskich koncepcji w zakresie tworzenia oprogramowania dla systemów regulowych, które mogą zostać rozszerzone na obszar logiki biznesowej i systemów sterowania.

Istotnymi osiągnięciami wnioskodawców w tej dziedzinie są narzędzia wspomagania projektowania oprogramowania: gKheops, Osiris, Mirella (implementujące koncepcje XTT), jak również prototypowe narzędzie QaRDE wspomagające konceptualne projektowanie oprogramowania bazujące na koncepcji ARD.

W dziedzinie systemów regulowych dotychczasowe wyniki zostały podsumowane w pracy [6] stanowiącej unikalną pozycję w tym obszarze; przedstawione tam rozwiązania konstytuują punkt wyjścia do rozszerzenia zaproponowanej metodologii na gruncie metod inżynierii oprogramowania. Istniejące narzędzia już w chwili obecnej realizują szereg istotnych komponentów pakietu narzędzi Hekate tj. QaRDE (edytor konceptualny), Mirella (edytor logiczny), Embedded Prolog Platform (EPP) [14] (środowisko uruchomieniowe).

2.3. Dlaczego projekt zdaniem autora powinien być finansowany?

Finansowanie projektu jest celowe z jednej strony ze względu na duże zapotrzebowanie na rozwijanie nowych, skuteczniejszych metod projektowania i efektywnej implementacji oprogramowania, z drugiej zaś na innowacyjność proponowanego podejścia. Finansowanie projektu umożliwi prezentację wypracowanych narzędzi oraz technologii na szerszym forum naukowym, włączając w to publikacje w czasopiśmie, oraz referaty na konferencjach międzynarodowych w kraju i za granicą.

Umożliwi ponadto przeprowadzenie praktycznych doświadczeń i prototypowych wdrożeń w dziedzinie inteligentnego sterowania z wykorzystaniem Platformy Mobilnego Robota Hexor oraz urządzeń klasy PDA.

Proponowane rozwiązanie może w przyszłości stać się podstawą do sformułowania standardu w dziedzinie wytwarzania oprogramowania. Tym bardziej, iż badania dotyczące wspomnianych zagadnień, włączając w to rozwiązania chronione prawem patentowym, a zatem znajdujące szerokie zastosowanie w przemyśle, są prowadzone w wiodących ośrodkach naukowych takich jak: NASA Goddard Space Flight Center [15] oraz NASA Ames Research Center [16]. Wątpliwości autorów nie budzi fakt, iż narzędzia opierające się na koncepcjach projektu Hekate, prędzej czy później zostaną opracowane w wiodących centrach rozwojowych w USA czy Europie o czym świadczą prowadzone tam badania [15, 16]. Zatem umożliwienie szybkiego rozwoju projektu w kraju, jest decyzją strategiczną.

2.4. Znaczenie wyników projektu dla rozwoju danej dziedziny i dyscypliny naukowej oraz rozwoju cywilizacyjnego.

Wyniki projektu będą mieć istotne znaczenia dla rozwoju nowych metod w inżynierii oprogramowania i praktycznych zastosowań inżynierii wiedzy. Proponuje się nakreślenie nowych standardów wytwarzania wysoce niezawodnego oprogramowania. Opierać się one będą na deklaratywnych i wizualnych zarazem metodach projektowania, które z kolei będą posiadały

odpowiadający im, dobrze zdefiniowany model logiczny. Standardy te przyczynią się do rozwoju cywilizacyjnego i wprowadzą nową, dotąd niespotykaną jakość, w procesie wytwarzania oprogramowania. Wzrost niezawodności oprogramowania zwiększy powszechność, ogólnodostępność i bezawaryjność technologii informatycznych.

Proponowane zmniejszenie luki semantycznej przyczyni się do redukcji kosztów oraz czasu wytwarzania oprogramowania.

2.5. Czy w przypadku pozytywnych wyników będą one mogły znaleźć praktyczne zastosowanie?

Zarówno teoretyczne, jak i praktyczne rezultaty projektu niewątpliwie będą mogły znaleźć zastosowania w następujących dziedzinach:

- przy rozwoju oprogramowania biznesowego,
- wspomagania konstruowania interfejsów HCI (ang. *Human-Computer Interaction*),
- a także w przemyśle dla aplikacji sterujących (szczególnie dla platform autonomicznych).

Proponowane przez wnioskodawców rozwiązania wzbudziły już zainteresowanie sektora komercyjnego branży IT. Spodziewane jest nawiązanie współpracy z firmami wytwarzającymi oprogramowanie. Współpraca taka umożliwi ugruntowanie utworzonych standardów, ich weryfikację w zastosowaniach komercyjnych, jak również realizację grantów zamawianych. Rozwiązania opracowane w ramach projektu mogą stać się podstawą do wnioskowania o patenty w kraju i za granicą.

3. Istniejący stan wiedzy w zakresie tematu badań (jaki oryginalny wkład wniesie rozwiązanie postawionego problemu do dorobku danej dyscypliny naukowej w kraju i na świecie, czy w kraju i na świecie jest to problem nowy czy kontynuowany i w jakim zakresie weryfikuje utarte poglądy i dotychczasowy stan wiedzy)

3.1 Jaki oryginalny wkład wniesie rozwiązanie postawionego problemu do dorobku danej dyscypliny naukowej w kraju i na świecie?

Oryginalny wkład wniesiony przez rozwiązania opracowane w ramach projektu będzie polegał między innymi na:

- skonstruowaniu prototypowego pakietu narzędzi CASE realizujących poniższy proces i metodologię wytwarzania oprogramowania,
- opracowaniu zintegrowanego procesu projektowania i implementacji oprogramowania, wzbogacającego istniejące metody inżynierii oprogramowania o elementy Inżynierii Wiedzy,
- przeprowadzeniu testowych implementacji oprogramowania, za pomocą w/w narzędzi, co pozwoli na ewaluację podejścia,
- zaproponowaniu metodologii tworzenia oprogramowania, rozszerzonej względem istniejących rozwiązań z Inżynierii Oprogramowania tj. UML.

3.2. Czy w kraju i na świecie jest to problem nowy czy kontynuowany?

Samo poszukiwanie nowych metod inżynierii programowania, oraz wskazywanie nowych zastosowań inżynierii wiedzy jest obszarem nieustających badań. Inżynieria Wiedzy znajduje zastosowanie wszędzie tam gdzie wymagane są wysokie poziomy niezawodności oprogramowania. Przykładami są zastosowania w dziedzinach eksploracji przestrzeni kosmicznej [15, 16], czy też sterowania przemysłowego (Kheops). Tym niemniej hybrydowe podejście proponowane w projekcie należy uznać za innowacyjne.

Koncepcja Hybrydowej Inżynierii Wiedzy opiera się między innymi na rozszerzeniu podejścia MVC (Model-View-Controller). Podejście to zostało zastosowane w klasycznym języku obiektowym Smalltalk. MVC wprowadziło deklaratywną klasyfikację elementów funkcjonalnych tworzonego systemu jako składniki trzech klas elementów tj. Model, View oraz Controller. Głównym atutem MVC jest separacja warstwy logiki biznesowej (Model) od prezentacji (View, Controller). W ostatniej dekadzie znalazło ono szerokie zastosowanie przy tworzeniu tzw. aplikacji biznesowych na platformach Java Enterprise Edition. W Hybrydowej Inżynierii Wiedzy proponuje

się, w podobny do MVC tj. deklaratywny sposób opisywania struktury systemu oraz deklaratywny opis logiki biznesowej, z której wynika funkcjonalność. Jednocześnie nie wyklucza się możliwości integracji deklaratywnych elementów typu Model z istniejącymi obiektowymi technologiami MVC, a w szczególności tymi opartymi o Java Enterprise Edition.

Wartym odnotowania jest fakt niejawnego wprowadzenia do nowych platform programistycznych (ang. *frameworks*) elementów deklaratywnego modelowania struktury oprogramowania. Przykładem może być rozwiązanie zastosowane w platformie Ruby On Rails. Określenie funkcjonalności oraz znaczenia modułów oprogramowania (klas), sformułowanych w języku obiektowym, odbywa się w sposób deklaratywny poprzez przyporządkowanie modułu do określonej kategorii. Kategorie implementują podejście MVC (patrz niżej) i w praktyce są reprezentowane przez odpowiednią hierarchię katalogów. Zaproponowane przez wnioskującego Zespół podejście rozszerza koncepcję MVC poza ramy zastosowania w Ruby on Rails.

Innym wątkiem jaki wyraźnie rysuje się w ostatnich latach w rozwijaniu oprogramowania biznesowego dużej skali jest tzw. podejście *Business Rules* [17]. W podejściu tym logikę systemu modeluje się w sposób deklaratywny, z użyciem reguł. We wstępnej fazie specyfikacji logiki systemu, reguły te są zapisywane w języku naturalnym. W kolejnych fazach następuje ich formalizacja i przetwarzanie przy pomocy istniejących powłok systemów ekspertowych, posiadających zaawansowane możliwości integracji z aplikacjami biznesowymi, w szczególności na platformie Java [18]. Przykładem takiej szeroko stosowanej powłoki jest środowisko Jess. Co jednak warto odnotowania, sam Jess nie wykorzystuje żadnych nowych osiągnięć inżynierii wiedzy. Jest on w praktyce jedynie reimplementacją, w języku Java, pochodzącego z lat 80 systemu CLIPS, i do pewnego stopnia anachronicznych na dzień dzisiejszy metodach reprezentacji i przetwarzania, a zwłaszcza sterowania wnioskowaniem. Z drugiej strony podejmowane próby tworzenia wizualnych środowisk programowania deklaratywnego (np. VisiRule, X-Pert) trudno jest uznać za obiecujące z uwagi na bardzo ograniczone możliwości.

Kolejne istotne wątki w inżynierii oprogramowania to proponowanie hierarchicznych podejść do rozwijania oprogramowania i użycie metod formalnych przy weryfikacji własności tworzonych systemów. Przykładem podejścia próbującego opisać proces tworzenia projektu w sposób bardziej całościowy może być MDA (ang. *Model Driven Architecture*) [19]. MDA, zaproponowane przez OMG (ang. *Object Management Group*) w roku 2001, jest próbą nadania formalnych ram procesowi projektowemu tak, by dzięki kolejnym uściśleniom modeli UML można było w efekcie końcowym otrzymać działający system. Wprowadza ono do procesu tworzenia oprogramowania cztery warstwy abstrakcji modeli: Computation Independent Model (CIM), Platform Independent Model (PIM), Platform Specific Model (PSM), oraz Implementation Specific Model (ISM).

Pierwsza warstwa (CIM), zwana czasem modelem biznesowym, zawiera tylko te elementy diagramów UML, które są niezbędne do zdefiniowania funkcjonalności biznesowej, oraz zakresu odpowiedzialności całego systemu. Kolejna warstwa PIM, zawiera ponadto logikę obliczeniową konieczną do realizacji zadań przedstawionych w modelu biznesowym. Szczegóły platformy systemowej oraz samej implementacji są definiowane w ramach dwóch ostatnich warstw PSM i ISM.

W podejściu MDA próbuje się formalizować przejścia pomiędzy tymi poziomami. W literaturze szczególnie nacisk kładzie się na możliwość zdefiniowania transformacji pomiędzy modelami PIM i PSM (patrz „MDA Guide Version 1.0.1”, OMG, 2003.). Nie ma gotowej odpowiedzi w jaki sposób taka transformacja ma wyglądać. W szczególności czasem musi być ona zdefiniowana i przeprowadzona ręcznie przez osobę projektującą system.

Warto też zauważyć, iż model powstały w oparciu o MDA jest obciążony wszystkimi problemami samego UML, jak choćby ciągle nieprecyzyjna semantyka języka UML [20]. Pomimo swoich niedostatków MDA stanowi ważny głos w dyskusji nad sposobem wykorzystania modeli w procesie projektowym, pokazując jak ważnym problemem jest nie tylko samo stworzenie modelu systemu, ale także jego przełożenie na produkt końcowy - gotową do uruchomienia aplikację.

W obszarze modelowania i analizy oprogramowania rozwijanych jest szereg specjalizowanych metod formalnych, takich jak Sieci Petriego [21], czy algebry procesów CCS, CSP, ACP, LOTOS i inne [22]. Pozwalają one na utworzenie modelu dynamiki systemu, a następnie formalną weryfikację jego zachowania względem zadanej specyfikacji. Modelowanie systemu z użyciem tych metod polega przede wszystkim na konstrukcji kolejnych uściśleń modelu tak, by w efekcie końcowym możliwie najpełniej odtworzyć zachowanie rzeczywistej aplikacji. Ścisła notacja oraz dobrze zdefiniowana semantyka pozwalają na formalne dowodzenie własności takich jak np. żywotność, bezpieczeństwo itp.

Precyzyjne, w wysokim stopniu sformalizowane podejście do modelowania systemów, stanowiące o sile wspomnianych metod, jest również ich słabością. Niewiele osób w codziennej praktyce potrafi metody te wykorzystywać. [23]. Tym ważniejsze jest szukanie sposobów

pełniejszego ich wykorzystania w szeroko rozumianej inżynierii oprogramowania, w ramach powszechnie uznanych, choć w mniejszym stopniu sformalizowanych, notacji i języków - takich jak choćby UML.

Z perspektywy inżynierii [24] wiedzy należy wskazać na pojawiające się w ostatnich latach próby łączenia deklaratywnego i logicznego programowania w języku Prolog z metodami obiektowymi. Dobrym przykładem jest tu platforma LogTalk (<http://www.logtalk.org>), dostarczająca mechanizmy obiektowe w Prologu, co potencjalnie umożliwi lepszą integrację z językami programowania obiektowego, takimi jak Java, czy C++. Są to jednak rozwiązania na wąsko zdefiniowanym poziomie narzędziowym, nie dostarczające kompletnych narzędzi.

3.3. W jakim zakresie rozwiązanie postawionego problemu weryfikuje utarte poglądy i dotychczasowy stan wiedzy?

Główne punkty wyjścia projektu, takie jak:

- użycie konceptualnych metod projektowania w inżynierii oprogramowania, wraz z możliwością analizy formalnej,
- użycie metod inżynierii wiedzy przy projektowaniu systemów opartych o wiedzę (np. systemów ekspertowych),

są ugruntowane, przynajmniej na poziomie koncepcji. Tym niemniej widać od kilku lat usilne próby przełamania tych klasycznych ugruntowanych podejść. Próbami takimi są wspomniane Business Rules, czy integracja oprogramowania typu Jess z aplikacjami biznesowymi. Coraz większy nacisk jest kładziony na technologie automatycznego generowania kodu na podstawie modelu konceptualnego i formalnej analizy struktury i dynamiki systemu.

Zdaniem Zespołu, oraz z perspektywy proponowanego podejścia Hekate, próby te należy uznać za niewystarczające i omijające istotę problemu. Proponowana integracja metod inżynierii wiedzy i oprogramowania, zdaniem Zespołu, doprowadzi do wypracowania metodologii bardziej efektywnych, oraz technologii i narzędzi przewyższających obecnie stosowane.

Wykorzystanie języków bardzo wysokiego poziomu (Prolog) i integracja występujących w nim mechanizmów ze współcześnie używanymi zaawansowanymi platformami takim jak Java pozwoli na zaproponowanie skuteczniejszych metod budowania oprogramowania biznesowego i ogólnego przeznaczenia. Stworzone w ramach Hekate rozwiązania służące do implementowania wbudowanych systemów inteligentnego sterowania pozwolą na użycie języków bardzo wysokiego poziomu do tworzenia takich systemów.

Podsumowując, należy uznać, iż rezultaty naukowe i inżynierskie projektu Hekate, pozwolą na zweryfikowanie istniejącego stanu wiedzy co do projektowania i implementacji oprogramowania dla szerokiej klasy systemów.

4. **Metodyka badań** (co stanowi podstawę naukowego warsztatu wnioskodawcy i jak zamierza rozwiązać postawiony problem, na czym będzie polegać analiza i opracowanie wyników badań, jakie urządzenia [aparatura] zostaną wykorzystane w badaniach, czy wnioskodawca ma do nich bezpośredni dostęp i umiejętność obsługi)

4.1 Co stanowi podstawę naukowego warsztatu wnioskodawcy?

Podstawy naukowego warsztatu wnioskodawców stanowią badania zapoczątkowane w [12] i dalej rozwijane w ramach projektów: Regulus, Mirella, Adder. Zaowocowały one między innymi praktycznymi implementacjami narzędzi:

- środowiska wspomagania projektowania przy pomocy XTT, analizy i implementacji systemów regułowych Mirella [10],
- środowiska wspomagania projektowania konceptualnego QaRDE,
- platformy dostarczającej komponent regułowy dla relacyjnych baz danych ReDaReS [11],

Wyżej wymienione narzędzia oparte są o koncepcje:

- wspomagania projektowania systemów regułowych, Tabular Trees [12],
- metody logicznej reprezentacji wiedzy dla regułowych systemów ekspertowych, Extended Tabular Trees (XTT) [10],
- metody konceptualnego prototypowania atrybutowych systemów regułowych, Attribute Relationship Diagrams (ARD) [6,9],

- platformy dla konstruowania wbudowanych systemów inteligentnego sterowania opartych o język Prolog, Embedded Prolog Platform (EPP) [14].

Zatem wnioskujący posiadają zarówno silne podstawy teoretyczne ([6], Projekt Regulus) jak i doświadczenie praktyczne (w/w implementacje) zarówno z dziedziny Inżynierii Wiedzy jak i Inżynierii Oprogramowania.

4.2. Jak zamierza wnioskodawca rozwiązać postawiony problem?

Jednym z istotnych wątków prowadzonych prac jest koncepcja atrybutowej logiki granularnej - języka do specyfikacji wiedzy deklaratywnej i równocześnie jej przetwarzania [25]. Opracowanie języka klasy VSAL (Variable Set Attributive Language) [6] stanowić będzie punkt wyjścia do realizacji projektu. W oparciu o zdefiniowany język określone zostaną metody reprezentacji wiedzy i jej przetwarzania. Opracowane zostaną też metody wizualizacji i komputerowego wspomaganie projektowania w postaci wspomnianego zestawu narzędzi.

Przewiduje się że rozwiązanie problemu zostanie wykonane w następujących etapach:

- przygotowanie serwisu WWW projektu opartego o system pracy grupowej Wiki,
- opracowanie zintegrowanego modelu logicznego reprezentacji, przetwarzania i sterowania przetwarzaniem wiedzy, w oparciu o Granularną Logikę Atrybutową,
- stworzenie pakietu prototypowych narzędzi komputerowego wspomaganie tworzenia oprogramowania,
- stworzenie platformy uruchomieniowej dla komponentów logiki biznesowej.
- stworzenie platformy uruchomieniowej dla wbudowanych systemów inteligentnego sterowania,
- prototypowe implementacje aplikacji testowych i ewaluacja stworzonych narzędzi.

4.3. Na czym będzie polegać analiza i opracowanie wyników badań?

Analiza będzie przeprowadzona na podstawie procesu testowania zaproponowanych narzędzi poprzez tworzenie prototypowych aplikacji. Rozpatruje się dwie klasy aplikacji tj. wysokopoziomową aplikację biznesową oraz inteligentny system sterowania Mobilną Platformą Hexor. Opracowanie wyników zostanie zaprezentowane w formie rozszerzonego raportu.

4.4. Jakie urządzenia zostaną wykorzystane w badaniach?

Do projektowania oraz rozwijania zaproponowanej metodologii oraz implementacji technologii zastosowane będą systemy komputerowe tj. stacje robocze i serwery. Wnioskodawcy będą korzystać głównie z oprogramowania typu FreeSoftware (wolnodostępnego) oraz OpenSource. Wykorzystywane będą między innymi otwarte implementacje systemu Unix, w tym GNU/Linux, a także zaawansowane kompilatory języka Prolog, takiej jak SWI Prolog, Yap Prolog, czy GNU Prolog. Przewiduje się również użycie oprogramowania IBM Rational będącego wiodącym na rynku pakietem udostępniającym metodologie UML.

Wyżej wymienione systemy komputerowe zostaną również wykorzystane do przetestowania zaimplementowanych technologii, w obrębie aplikacji ogólnego przeznaczenia oraz biznesowych. W celu implementacji i przetestowania aplikacji sterujących oraz aplikacji dla systemów wbudowanych wykorzystane będą: Platforma Mobilnego Robota Hexor oraz urządzenia typu PDA.

4.5. Czy wnioskodawca ma bezpośredni dostęp i umiejętność obsługi wspomnianych urządzeń?

Wnioskodawcy mają dostęp do wyżej wspomnianych systemów komputerowych oraz oprogramowania w Katedrze Automatyki AGH. Mają również bogate doświadczenia w zakresie obsługi i wykorzystania tego typu sprzętu i oprogramowania. W roku 2006 KA AGH zakupiła jednego robota Hexor, który posłużył dotychczas do wstępnej ewaluacji niektórych koncepcji opisanych w niniejszym wniosku. Planowane są zakupy kolejnych robotów Hexor oraz urządzeń typu PDA.

Zarówno w przypadku systemów komputerowych, robota Hexor, jak i urządzeń PDA wnioskujący posiadają umiejętności ich obsługi i programowania. Wnioskujący Zespół ma również

bogate doświadczenia praktyczne i wiedzę teoretyczną w dziedzinach Inżynierii Wiedzy, Inżynierii Oprogramowania i Relacyjnych Baz Danych.

5. Wymierny, udokumentowany efekt podjętego problemu (zakładany sposób przekazu i upowszechnienia wyników - publikacje naukowe oraz referaty na konferencjach w kraju i za granicą, monografie naukowe, rozprawy doktorskie i habilitacyjne, nowe metody i urządzenia badawcze)

5.1 Zakładany sposób przekazu i upowszechniania wyników.

Wyniki naukowe i praktyczne projektu, będą prezentowane na konferencjach międzynarodowych za granicą (FLAIRS USA) i w kraju (CMS/MSK). Planowane jest publikowanie najważniejszych koncepcji w czasopismach z tzw. Listy Filadelfijskiej. Stan i postępy prac będą na bieżąco publikowane w serwisie internetowych projektu, który został uruchomiony w czerwcu 2006 roku pod adresem <http://hekate.ia.agh.edu.pl>. Zakłada się, iż publikacje na konferencjach i w czasopismach będą głównie w języku angielskim, podczas gdy serwis internetowy będzie dwujęzyczny, polsko-angielski.

5.2. Monografie naukowe.

Na planowanym etapie nie przewiduje się sporządzenia monografii naukowych. Wyniki będą publikowane na bieżąco w serwisie Wiki.

5.3. Rozprawy doktorskie i habilitacyjne.

Planowane jest zrealizowanie jednej pracy doktorskiej, oraz przygotowanie drugiej; a także zebranie materiałów do dwóch prac habilitacyjnych.

5.4. Nowe metody i urządzenia badawcze.

Planowana jest między innymi:

- implementacja oprogramowania sterującego opartego o paradygmaty Inżynierii Wiedzy na Platformie Mobilnego Robota Hexor; platforma ta będzie mogła posłużyć jako punkt wyjście do badań nad autonomicznymi systemami mobilnymi, przewiduje się udzielenie licencji na w/w oprogramowanie,
- opracowanie uniwersalnej platformy dostarczającej regułowego środowiska programowania dla systemów wbudowanych,
- dostarczenie metody integracji deklaratywnego programowania w logice z aplikacjami biznesowymi.

W przypadku oprogramowania dla systemów wbudowanych zostaną skonstruowane i zaprezentowane funkcjonujące prototypy urządzeń.

6. Wykaz najważniejszej literatury dotyczącej problematyki wniosku (maks. 25 pozycji)

- [1] Sommerville I., Software Engineering, 7th ed., Pearson Education Limited, 2004.
- [2] Hopgood A. A., Intelligent Systems for Engineers and Scientists, CRC Press, 2001.
- [3] Russell S., Norvig P., Artificial Intelligence: A Modern Approach, Prentice-Hall, 2nd ed., 2002.
- [4] Negnevitsky M., Artificial Intelligence. A Guide to Intelligent Systems, Addison-Wesley, 2002.
- [5] Liebowitz J., The Handbook of Applied Expert Systems, CRC Press, 1998.
- [6] Ligęza A.: Logical Foundations for Rule-Based Systems. Springer-Verlag, Berlin, Heidelberg, 2006. Studies in Computational Intelligence, Vol. 11.
- [7] Bratko I., Prolog Programming for Artificial Intelligence, Addison Wesley, 3rd ed., 2000.
- [8] A. Newell. The Knowledge Level, Artificial Intelligence, 18(1):87-127, 1982.
- [9] Nalepa G. J., Ligęza A.: Conceptual modelling and automated implementation of rule-based systems, W: Software engineering : evolution and emerging technologies / eds. Krzysztof Zieliński, Tomasz Szmuc. Amsterdam : IOS Press, 2005. (Frontiers in Artificial Intelligence and Applications ; vol. 130). S. 330-340.

- [10] Nalepa G. J., Ligęza A.: A visual edition tool for design and verification of knowledge in rule-based systems, *Systems Science*, 2005 vol. 31 no. 3 s. 103-109.
- [11] Wojnicki I. , Ligęza A., Jelly View - A Technology for Arbitrarily Advanced Queries within RDBMS, SAC 2005, The 20th Annual ACM Symposium on Applied Computing Santa Fe, New Mexico, 1317 Marzec 2005.
- [12] Tab-Trees: a CASE tool for the design of extended tabular systems, W: Database and expert systems applications : 12th international conference, DEXA 2001 : Munich, September 3-5, 2001 : proceedings / eds. Heinrich C. Mayr [et al.], Springer, 2001. (Lecture Notes in Computer Science ; 2113). S. 422-431.
- [13] Nalepa G. J., Ligęza A.: Prolog-based analysis of tabular rule-based systems with the XTT approach, W: FLAIRS 2006 : proceedings of the nineteenth international Florida Artificial Intelligence Research Society conference, / eds. Geoffrey C. J. Sutcliffe, Randy G. Goebel ; FLAIRS, Menlo Park : AAI Press, 2006. S. 426-431.
- [14] Nalepa G. J., Zięcik P.: Integrated embedded prolog platform for rule-based control systems, W: MIXDES 2006 : MIXed DESign of integrated circuits and systems : proceedings of the international conference : Gdynia, Poland 22-24 June 2006 / ed. Andrzej Napieralski., Technical University Lodz. Department of Microelectronics and Computer Science, 2006, S. 716-721.
- [15] Rash J. L., Hinchey M. J., Rouff C. A., Gracanin D., Erickson J., A Requirements-Based Programming Approach to Developing a NASA Autonomous Ground Control System, *Artificial Intelligence Review*, Springer, 2005/11.
- [16] Morris R., AI for Autonomy in Space Exploration: Current Capabilities and Future Challenges - Invited speech, FLAIRS 2006 : proceedings of the nineteenth international Florida Artificial Intelligence Research Society conference, / eds. Geoffrey C. J. Sutcliffe, Randy G. Goebel ; FLAIRS, Menlo Park : AAI Press, 2006. S. 13-14.
- [17] Ross R. G., Principles of the Business Rule Approach, Addison-Wesley Professional, 2003.
- [18] Friedman-Hill E., Jess in Action: Java Rule-Based Systems, Manning Publications, 2003.
- [19] Kleppe A., Warmer J., Bast W., MDA Explained: The Model Driven Architecture - Practice and Promise, Addison-Wesley, 2003
- [20] Thomas D., MDA: Revenge of Modelers or UML Utopia, *IEEE Software*, 2004.
- [21] Brauer W., Rozenberg G., Salomaa A. (Eds.), Petri Nets, An Introduction, W. Reisig, EATCS, Monographs on Theoretical Computer Science, Springer Verlag, Berlin, 1985.
- [22] Fencott C., Formal Methods for Concurrency, International Thomson Computer Press, 1996.
- [23] Heitmeyer C., On the Need for Practical Formal Methods, *Lecture Notes in Computer Science*, 1998.
- [24] Menzies T. J., Applications of Abduction: Knowledge Level Modeling, *International Journal of Human Computer Studies*, volume 45: 305-355, 1996.
- [25] Ligęza A., Fuster Parra P.: A granular Attribute logic for rule-based systems management within extended tabular trees. W: R. Trappl (ed.): *Cybernetics and Systems*, Vol. 2, 761-766. Proceedings of the Eighteenth European Meeting on Cybernetics and Systems Research, Vienna, 2006.

E. HARMONOGRAM WYKONANIA PROJEKTU BADAWCZEGO – PLAN ZADAŃ

Lp.	Nazwa zadania badawczego	Termin rozpoczęcia*/ ----- zakończenia*/ zadania	Przewidywane koszty (zł)
1	2	3	4
1.	Przygotowanie serwisu WWW projektu opartego o system pracy grupowej Wiki.	1-2	10 000
2.	Opracowanie zintegrowanego modelu logicznego reprezentacji, przetwarzania i sterowania przetwarzaniem wiedzy.	3-9	95 000
3.	Utrzymywanie i aktualizacja serwisu WWW, prezentującego wyniki projektu.	3-24	25 000
4.	Adaptacja metodologii UML na potrzeby modelowania systemów z baz wiedzy.	6-9	35 000
5.	Stworzenie pakietu prototypowych narzędzi komputerowego wspomaganie tworzenia oprogramowania.	9-20	90 000
6.	Stworzenie platformy uruchomieniowej dla komponentów logiki biznesowej.	14-20	30 000
7.	Stworzenie platformy uruchomieniowej dla wbudowanych systemw inteligentnego sterowania.	16-20	30 000
8.	Prototypowe implementacje aplikacji testowych i ewaluacja stworzonych narzędzi.	20-24	20 000
9.	Analiza jakościowa i weryfikacja otrzymanych wyników.	22-24	5 500
RAZEM			340 500

*/ Podać liczbę miesięcy liczonych od faktycznego terminu rozpoczęcia projektu

UWAGA

Nie są zadaniami badawczymi czynności techniczne służące wykonaniu zadania np. zakup aparatury i materiałów, opracowanie raportów itp.

F. KOSZTORYS PROJEKTU BADAWCZEGO

1. Poszczególne pozycje kosztorysu w cenach bieżących (zł)

Lp.	Treść	Planowane koszty w roku		
		2007	2008	Razem
1	2	3	4	5
1	Koszty bezpośrednie	155 000	115 000	270 000
	w tym:			
	1/ Wynagrodzenia z pochodnymi	80 000	75 000	155 000
	2/ Inne koszty realizacji projektu (łącznie z kosztem zakupu lub wytworzenia aparatury naukowo - badawczej)	75 000	40 000	115 000
2	Koszty pośrednie	36 000	34 500	70 500
3	Koszty ogółem finansowane ze środków finansowych na naukę (1+2)	191 000	149 500	340 500
4	Koszty ogółem finansowane z innych źródeł niż środki finansowe na naukę	0	0	0
5	Koszty ogółem (3+4)	191 000	149 500	340 500

2. Kalkulacja poszczególnych pozycji kosztorysu

1) Wynagrodzenia z pochodnymi: 155 000 zł

Liczba osób przewidzianych do udziału w realizacji projektu: 12

Liczba osobomiesięcy

ogółem:	160
kierownika projektu:	24
wykonawców projektu:	136

2) Opis planowanych zakupów lub wytworzenia aparatury naukowo - badawczej

(podać nazwę zakupu, planowany przewidywany koszt, planowany miesiąc zakupu lub wytworzenia liczony od rozpoczęcia realizacji projektu oraz merytoryczne uzasadnienie)

- Rozszerzona wersja Roboty Mobilnego Hexor wraz z dodatkowym wyposażeniem, szacowany koszt: **18 000 zł**, planowany miesiąc zakupu: **3 i 9**, niezbędne do weryfikacji platformy uruchomieniowej dla inteligentnego systemu sterowania (patrz Opis Projektu Badawczego)
- Urządzenia i platformy, wraz z oprogramowaniem i licencjami, w tym między innymi do: wspomaganie projektowania i prototypowania oprogramowania i systemów przenośnych oraz wbudowanych, włączając w to urządzenia klasy PDA, oparte o procesory FreeScale, ARM, AVR, szacowany koszt: **17 000 zł**, planowany miesiąc zakupu: **3 i 9**, niezbędne do weryfikacji platformy uruchomieniowej dla inteligentnego systemu sterowania (patrz Opis Projektu Badawczego)

3) Uzasadnienie wysokości planowanych innych kosztów realizacji projektu

(wymienić rodzaj kosztów, wysokość oraz ich powiązanie z planem zadań projektu)

- wyjazdy krajowe, koszt: **18 000 zł**, zadania: **1-9**, uzasadnienie: prezentacja rezultatów badań na prestiżowych konferencjach naukowych w kraju
- wyjazdy zagraniczne, koszt: **60 000 zł**, zadania: **1-9**, uzasadnienie: prezentacja rezultatów badań na prestiżowych konferencjach naukowych za granicą
- materiały eksploatacyjne: koszt: **2 000 zł**, zadania: **3-9**

G. OŚWIADCZENIA I PODPISY

1. Oświadczam, że zapoznałem się z wnioskiem o finansowanie projektu badawczego pt.:
Hybrydowa Inżynieria Wiedzy: Zintegrowane narzędzia komputerowe do efektywnego wytwarzania niezawodnego oprogramowania w oparciu o deklaratywny model logiczny. (HEKATE)
2. W przypadku przyjęcia projektu badawczego do finansowania jednostka zobowiązuje się do:
 - włączenia projektu do planu zadaniowo-finansowego jednostki,
 - udostępnienia pomieszczeń, aparatury i obsługi administracyjno-finansowej,
 - zatrudnienia pracowników niezbędnych do realizacji projektu na podstawie uzgodnionej z kierownikiem projektu i właściwymi wykonawcami formy zatrudnienia (mianowanie, umowa o pracę, umowa o dzieło, umowa zlecenia),
 - sprawowania nadzoru nad realizacją projektu i prawidłowością wydatkowania środków finansowych
3. Oświadczam, że jednostka nie może zapewnić dostępu do urządzeń wymienionych w wykazie aparatury w części F ust. 2 pkt 2
4. Oświadczam, że projekt obejmuje badania¹⁾:
 - ~~1) wymagające doświadczeń na zwierzętach¹⁾~~
 - ~~2) nad gatunkami chronionymi lub na obszarach objętych ochroną¹⁾~~
 - ~~3) nad organizmami genetycznie modyfikowanymi lub z zastosowaniem takich organizmów¹⁾~~

Do wniosku dołączono dokumenty określone w § 31 ust. 5, 6 i 7¹⁾ rozporządzenia Ministra Nauki i Informatyzacji z dnia 4 sierpnia 2005 r. w sprawie kryteriów i trybu przyznawania i rozliczania środków finansowych na naukę

Oświadczam, że zgodnie z moją wiedzą przygotowany wniosek o finansowanie projektu badawczego nie narusza praw osób trzecich

I. INFORMACJE O OSOBIE ODPOWIEDZIALNEJ ZA SPORZĄDZENIE WNIOSKU

Imię i nazwisko, telefon, fax, e-mail: **Antoni Ligęza, 012 634 15 68, ligeza@agh.edu.pl**

Wniosek sporządzono (miejscowość, data): Kraków, 24.07.2006

Pieczęć jednostki	Kierownik jednostki (Rektor/Dyrektor)	Główny księgowy/Kwestor	Kierownik projektu
Data	podpis i pieczęć	podpis i pieczęć	podpis

UWAGI

5. W przypadku projektu promotorskiego należy dołączyć dokumenty określone w § 43 ust. 2 rozporządzenia Ministra Nauki i Informatyzacji z dnia 4 sierpnia 2005 r. w sprawie kryteriów i trybu przyznawania i rozliczania środków finansowych na naukę
 6. W przypadku wniosku składanego przez jednostkę naukową, która nie otrzymuje dotacji na działalność statutową do wniosku dołącza się dokumenty wymienione w § 31 ust. 3 i 4 rozporządzenia, o którym mowa w uwadze 1
- ¹⁾ Niepotrzebne skreślić
- ²⁾ W przypadku projektu habilitacyjnego należy podać informacje pozwalające na ocenę dorobku naukowego kierownika projektu powstałego w okresie od uzyskania stopnia naukowego doktora