

PIOTR FILIPKOWSKI

Instytut Informatyki i Gospodarki Cyfrowej
Szkoła Główna Handlowa w Warszawie

MICHAŁ HORODELSKI

Instytut Matematyki i Informatyki
Katolicki Uniwersytet Lubelski Jana Pawła II

Zastosowanie KBS do szybkiej identyfikacji częstotliwości z EEG¹

1. Wstęp

Ocena stanu zdrowia psychicznego jest niemniej ważna niż ocena stanu zdrowia fizjologicznego człowieka². Dzięki zaawansowanej technologii elektroniki medycznej możliwe staje się spojrzenie głębiej w stan umysłu, a nawet analiza stanu psychofizjologicznego człowieka. Zaawansowane urządzenia EEG odbierają sygnały niskiej mocy i dostarczają wysokiej jakości obraz numeryczny fal elektromagnetycznych generowanych przez ludzki mózg³. Sprawne i skuteczne używanie tego typu urządzeń wymaga od terapeuty olbrzymiej wiedzy z zakresu neuropsychologii w celu prawidłowej interpretacji odbieranych sygnałów. Wiedzę tę można jednak skodyfikować i zaszyć w algorytmach komputerowych. Pomocą służy tu inżynieria wiedzy oraz systemy

¹ Autorzy dziękują prof. A. Janickiemu z Wojskowego Instytutu Medycyny Lotniczej za dostęp do wiedzy i wprowadzenie w obszar badań z zakresu algorytmizacji procesów decyzyjnych w warunkach braku wiedzy *a priori* przy ostrych uwarunkowaniach czasowych w stresie, wspomagania decyzji wielokryterialnych, adaptacyjnej syntezy struktury hierarchicznych systemów informacyjnych z bazami wiedzy, tu m.in. w obszarze biofeedbacku. Autorzy dziękują także prof. A. Kobylińskiemu za umożliwienie dostępu do Wirtualnego Laboratorium LabIIGC™ Szkoły Głównej Handlowej w Warszawie i testowania KBS z uwzględnieniem rozwiązań sieciowych, będących kolejnym etapem prac rozwojowych. Podziękowania składają również G. Przybylskiemu – prezesowi GP Inter-Solution Sp. z o.o. – za udostępnienie zaawansowanej aparatury pomiarowej.

² P. Jaśkowski, *Zarys psychofizjologii*, Wyższa Szkoła Finansów i Zarządzania w Warszawie, Warszawa 2004.

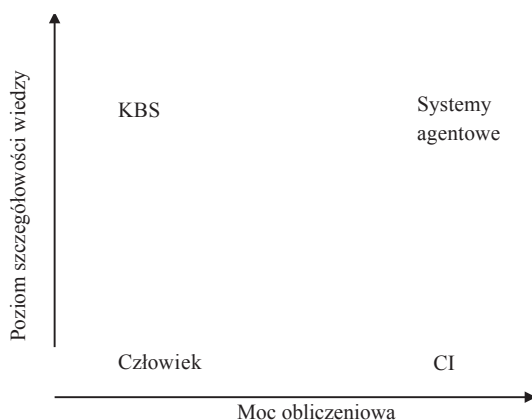
³ W. Błasiak, M. Godlewska, R. Rosiek, D. Wcisło, *Nowe technologie w badaniach edukacyjnych*, w: *Człowiek, media, edukacja*, red. J. Morbitzera, E. Musiał, Katedra Technologii i Mediów Edukacyjnych, Uniwersytet Pedagogiczny, Kraków 2013.

oparte na wiedzy (ang. *knowledge base system* – KBS). Autorzy w celu przyspieszenia procesu oceny stanu zdrowia psychofizjologicznego stworzyli program komputerowy, który wykorzystuje bazę wiedzy do oceny stanu psychofizjologicznego osoby badanej w procesie analizy biofeedbacku.

2. Wspomaganie decyzji a KBS

Wspomaganie decyzji jest nieodzownym procesem w warunkach ostrego uwarunkowania czasowego. Ze względu na to, że współczesna technika obliczeniowa umożliwia odciążenie człowieka od żmudnych, powtarzalnych prac umysłowych, możliwe jest przesunięcie jego roli z wykonawczej do kontrolnej. Dzięki narzędziom wspomagania decyzji można dbać o wyższą jakość i/lub niższy koszt wypracowanych decyzji.

Systemy z bazami wiedzy wydzielonymi od pozostałych modułów programu nazywa się systemami opartymi na bazie wiedzy (KBS). Systemy takie często są wykorzystywane jako systemy wspomagające podejmowanie decyzji (ang. *decision support systems*)⁴. Dzięki wykorzystaniu tsibuntu było możliwe opracowanie aplikacji pełniącej funkcję KBS z uwzględnieniem przyszłych rozszerzeń chmurzastych (np. w LabIIIGCTTM), zwiększających potencjał obliczeniowy do poziomu złożoności charakterystycznego dla systemów agentowych.



Rysunek 1. Efektywne rozwiązania

Źródło: opracowanie własne.

⁴ *Inteligentne systemy w zarządzaniu – teoria i praktyka*, red. J. S. Zieliński, Wydawnictwo Naukowe PWN, Warszawa 2000.

Opracowany przez autorów KBS 2014 jest swego rodzaju inteligentnym systemem opartym na wiedzy, którego potencjał (w zakresie rozwiązywania danego problemu) tkwi w zakodowanej w nim wiedzy, a nie w formalizmie i schematach wnioskowania, których ten program używa. Można to wyrazić w postaci stwierdzenia „more knowledge less search” („im pełniejsza jest wiedza, tym szybciej uzyskuje się rozwiązanie”). Problem z posiadaniem pełnej wiedzy tkwi więc w bazie wiedzy, a nie w sposobie realizacji procesu wnioskowania⁵. W przypadku braku wiedzy istotną rolę zaczynają odgrywać systemy inteligencji obliczeniowej (ang. *computational intelligence* – CI)⁶, które są przewidziane do zastosowania w kolejnych wersjach opracowanego oprogramowania.

3. Środowisko systemowe tsibuntu

Autorzy publikacji zdecydowali, iż rozwiązanie problemu postawionego w pracy zostanie zrealizowane za pomocą aplikacji umieszczonej na skonfigurowanej platformie modelowania i symulacji⁷. Taki model środowiska pozwala uruchomić aplikację na prawie dowolnie wybranym, typowym komputerze PC czy podręcznym laptopie, a także w środowisku chmurzastym.

System tsibuntu⁸ jest dystrybucją Linuxa opartą na systemie Ubuntu w wersji 10.04⁹, skompilowaną do postaci platformy modelowania i symulacji LabTSI®. System ten pozwala na prowadzenie obliczeń rozproszonych przy wykorzystaniu mechanizmu PVM i paradygmatu programowania klient–serwer¹⁰. Ponadto, oferuje niezależne środowisko

⁵ J.J. Mulawka, *Systemy ekspertowe*, WNT, Warszawa 1996.

⁶ P. Filipkowski, M. Horodelski, A. Kozłowski, P. Kurowski, J. Lasyk, W. Mruk, D. Oleszczuk, *Technologie społeczeństwa informacyjnego*, cz. 1, „Przegląd Techniczny” 2012, nr 22; P. Filipkowski, M. Horodelski, A. Kozłowski, P. Kurowski, J. Lasyk, W. Mruk, D. Oleszczuk, *Technologie społeczeństwa informacyjnego*, cz. 2, „Przegląd Techniczny” 2012, nr 23–24.

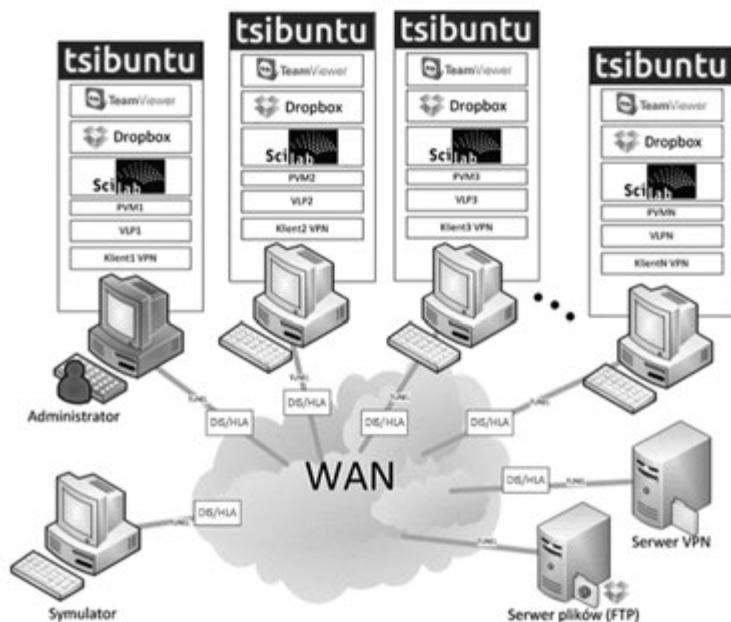
⁷ P. Filipkowski, M. Horodelski, *Zastosowanie rozproszonej platformy modelowania i symulacji do analizy informacji medycznych na bieżąco*, „Roczniki” Kolegium Analiz Ekonomicznych, z. 35, Oficyna Wydawnicza SGH, Warszawa 2014.

⁸ System ten powstał w Katedrze Technologii Społeczeństwa Informacyjnego Katolickiego Uniwersytetu Lubelskiego Jana Pawła II. Obecnie jest rozwijany przez członków stowarzyszenia Europejskie Centrum Technologii Społeczeństwa Informacyjnego w Białymstoku przy współpracy z Wojskowym Instytutem Medycyny Lotniczej w Warszawie. Ponadto, został włączony jako jeden z podstawowych elementów do Wirtualnego Laboratorium Instytutu Informatyki i Gospodarki Cyfrowej Szkoły Głównej Handlowej w Warszawie LabIiGC™.

⁹ P. Filipkowski, M. Kasperk, J. Nowakowski, *System operacyjny Linux tsibuntu live*, w: *LabTSI™ – platforma modelowania i symulacji*, red. A. Janicki, Wydawnictwo KUL, Lublin 2011.

¹⁰ P. Filipkowski, M. Kasperk, J. Nowakowski, *Wirtualna Maszyna Równoległa*, w: *LabTSI™...*, op.cit.

uruchomieniowe dla aplikacji typu KBS 2014. Dzięki temu można uruchomić aplikację w niezależnym systemie na komputerze, nie ingerując w jego system i opierając się jedynie na jego stronie sprzętowej. Aktualną konfigurację platformy modelowania i symulacji LabTSI® zaprezentowano na rysunku 2.



Rysunek 2. Tsibuntu 2013 w LabTSI®

Źródło: materiały wewnętrzne ECTSI.

Do uruchomienia systemu KBS 2014 wystarczy jeden węzeł. Skalowanie o większą liczbę węzłów umożliwia badanie tego typu systemów opartych na wiedzy oraz rozwój ich w kierunku przetwarzania rozproszonego. Uzyskana innowacyjność opracowanego rozwiązania tkwi w syntezy jego mobilności (wersja LIVE), powszechności i efektywności środowiska programowania (SCILAB oraz Scilab GUI) i skalowalności (dzięki PVM moc obliczeniowa może być dynamicznie powiększana w zależności od potrzeb). Ponadto, stosowane technologie są typu *Open Source* oraz objęto je licencją GNU-GPL, a wiedza zgromadzona w aplikacji jest tworzona wyłącznie na podstawie przeprowadzonych badań¹¹ w Wojskowym Instytucie Medycyny Lotniczej w Warszawie.

¹¹ W ramach projektu wdrożeniowego „Stworzenie aplikacji do interpretacji danych z urządzenia FlexComp Infinity V.6.1”, finansowanego przez PARP.

Scilab GUI (w skr. SGUI) to środowisko budowania wizualnych aplikacji desktopowych, może dostarczać rozwiązań biznesowych, a także być wsparciem przy pracy naukowej. Badacz, znając podstawy GUI i Scilaba, może w szybki sposób opracować własne symulacje wraz z wizualizacją rozwiązań problemów naukowych w formie aplikacji okienkowej. W opracowaniu KBS 2014 wykorzystano guibuilder w wersji 2.2 i Scilab 5.3.3.

Środowisko Scilab jest wysmienitym narzędziem inżynierskim, darmowym oprogramowaniem, które może zainstalować i używać wedle uznania każdy zainteresowany. Scilab, darmowa alternatywa dla Matlaba, udostępnia zestaw podstawowych operacji do działania na macierzach, takich jak m.in.: generowanie macierzy specjalnych, algebra macierzy, rozkłady macierzy, rozwiązywanie równań liniowych i nieliniowych, wyszukiwanie przybliżonych rozwiązań oraz wizualizacja danych w postaci wykresów, a także implementacja dynamicznych aspektów przepływu sterowania, w tym elementów decyzyjnych. Scilab jest podstawowym składnikiem systemu tsibuntu, ma budowę modułową i umożliwia rozszerzanie podstawowej instalacji o pakiety ATOMS, m.in. pakiet Scilab GUI¹².

4. System KBS 2014

Opracowany *knowledge base system* jest to system, którego głównym celem jest wspomaganie decyzji operatora urządzeń EEG w ocenie stanu psychofizjologicznego osoby badanej, np. pilota. Dzięki zgromadzonej wiedzy na temat przebiegów częstotliwości aktywności mózgu i dostępnym narzędziom inżynierii wiedzy funkcjonujących w autorskiej konfiguracji systemowej LabTSI[®] powstała innowacyjna aplikacja desktopowa KBS 2014. Aplikacja została zintegrowana z systemem tsibuntu LIVE. W systemie KBS 2014 możemy wyróżnić dwa ekrany startowy i ekran interpretera.

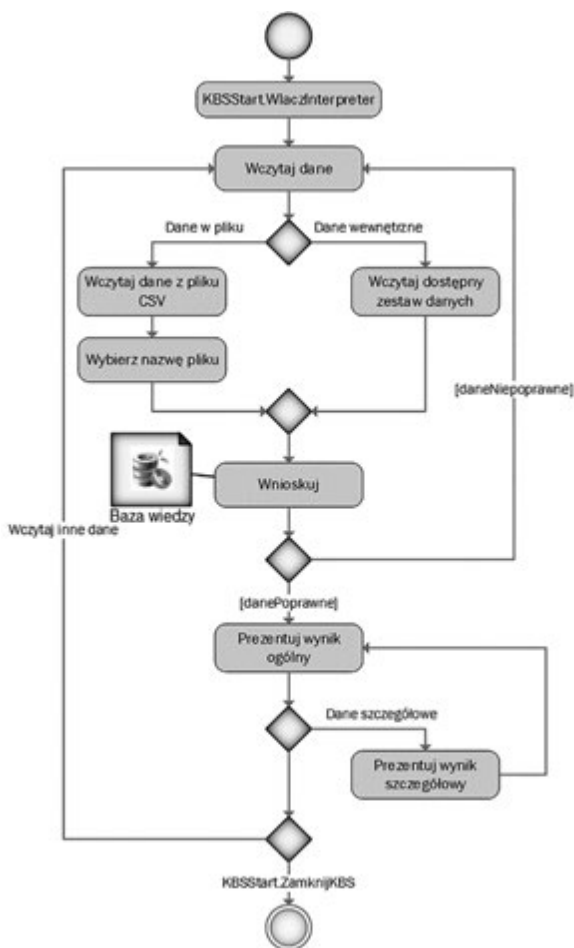
Intuicyjną obsługę ułatwia górne menu, dzięki któremu możliwe jest uruchomienie najważniejszych akcji programu, takich jak: „uruchom interpreter”, „wczytaj dane”, „interpretuj dane”, „wyczyść dane”. System KBS 2014 w cyklu swojego działania przechodzi przez kilka ważnych stanów. Stany oraz przejścia pomiędzy nimi przedstawiono przy użyciu języka UML na diagramie maszyny stanowej¹³, przedstawionej na rysunku 4.

¹² M. Horodelski, *Narzędzie modelowania i symulacji Scilab*, w: *LabTSI™...*, op.cit.

¹³ S. Wrycza, B. Marcinkowski, K. Wyrzykowski, *Język UML 2.0 w modelowaniu systemów informatycznych*, Helion, Gliwice 2006.

System po uruchomieniu jest w stanie wyświetlania ekranu startowego. Następnie może przejść do stanu wczytywania danych lub od razu uaktywnionego interpretera. Po wczytaniu danych (również z chmury) system także przechodzi do stanu aktywnego interpretera. W każdej chwili, poza stanami „wczytywanie danych z pliku”, „prezentacja interpretacji szczegółowej” oraz „informacje o autorach”, użytkownik może wyłączyć system.

Głównym i podstawowym przypadkiem użycia systemu KBS jest wnioskowanie, którego wynikiem jest interpretacja danych (interpretuj dane). Scenariusz główny tego przypadku dobrze prezentuje diagram aktywności na rysunku 5.



Rysunek 5. Diagram czynności dla przypadku użycia „interpretuj dane”

Źródło: opracowanie własne.

Aktorami uczestniczącymi w interakcji z systemem są użytkownik oraz urządzenie psychofizjologiczne, które dostarcza danych do interpretacji. Użytkownik po wczytaniu danych i przeprowadzeniu procesu wnioskowania wprost może wyświetlić wynik interpretacji w sposób dwupoziomowy – jako interpretacja ogólna oraz interpretacja szczegółowa dla konkretnych czujników urządzenia.

5. Ocena jakości metodą *model checking*

Do zbadania zgodności (poprawności) gotowego systemu KBS z nieformalnym opisem wymagań wykorzystano weryfikację abstrakcyjną – metodę *model checking*. Ze względu na to, że weryfikowany jest model systemu, a nie sam system, jest to weryfikacja abstrakcyjna¹⁴. Uwagę skupiono głównie na szczególnie ważnym ze względu na proces wnioskowania komponencie – podsystemie interpreter aktywny. W tym podejściu możliwa jest częściowa automatyzacja procesu weryfikacji, ale wymagana jest także ingerencja eksperta, który zbuduje model systemu oraz stworzy formalny opis wymagań systemowych na potrzeby weryfikacji¹⁵.

Metoda *model checking* jest formalną weryfikacją systemu, pozwalającą na sprawdzenie, czy jest on zgodny z jego założeniami. Metoda ta może być stosowana przy małych i średnich systemach. Polega na stworzeniu modelu systemu (modelu Kripke¹⁶), odzwierciedlającego jego stany, przejścia, ponadto opracowaniu formalnej reprezentacji specyfikacji systemu (w postaci formuł specyficznych dla CTL¹⁷) oraz automatycznym dowodzeniu zgodności modelu z wymaganiami, a więc opracowanym zbiorem formuł. W wyniku otrzymujemy informację o spełnieniu wymagań lub informację z przykładem, który pokazuje, iż wymagania nie są spełnione¹⁸.

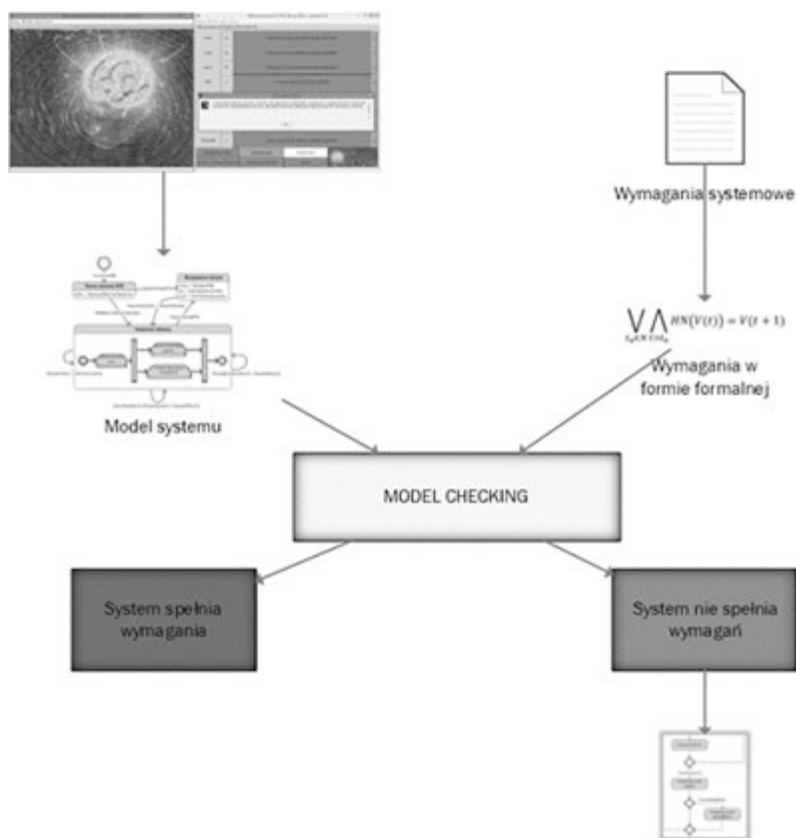
¹⁴ E. M. Clarke, *The Birth of Model Checking*, w: *25 Years of Model Checking*, red. O. Grumberg, H. Veith, „Lecture Notes in Computer Science”, cz. 5000, Springer, Berlin–Heidelberg 2008, s. 1–26.

¹⁵ E. M. Clarke, *Model Checking*, w: *Handbook of Automated Reasoning*, red. A. Robinson, A. Voronkov, Elsevier BV, Amsterdam 2001, s. 1635–1790.

¹⁶ S. Kripke, *Semantical Considerations on Modal Logic*, „Acta Philosophica Fennica” 1963, vol. 16, s. 83–94.

¹⁷ E. A. Emerson, E. M. Clarke, *Using branching time logic to synthesize synchronization skeletons*, „Science of Computer Programming” 1982, vol. 2, s. 241–266.

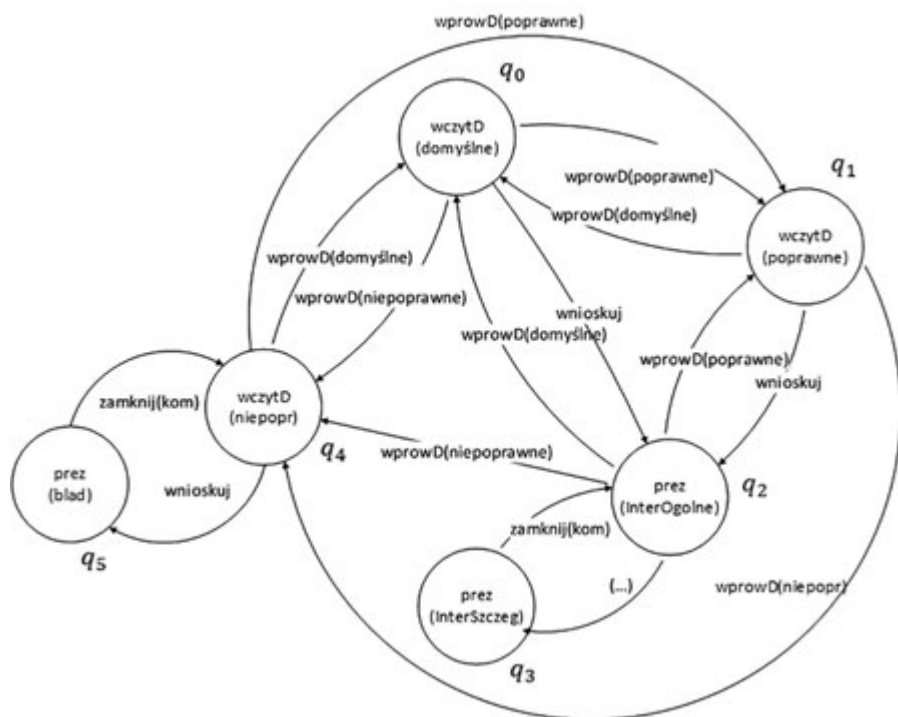
¹⁸ E. M. Clarke, *Model...*, op.cit.



Rysunek 6. Schemat weryfikacji systemu KBS za pomocą metody *model checking*

Źródło: opracowanie własne.

Autorzy, wykorzystując wiedzę ekspercką, zbudowali model Kripke podsystemu interpreter aktywny (w skr. MKpIA, prezentowany na rysunku 7), który następnie zostanie wykorzystany w procesie weryfikacji. Jest to rozszerzony model Kripke o dodatkowe nazwy dla przejść, zawierający informację o stanach, przejściach pomiędzy stanami oraz predykatach (predykat jest rozumiany w niniejszej pracy jako funkcja zdaniowa lub zdanie) prawdziwych w odpowiednich stanach. Powiązanie predykatu ze stanem jest równoznaczne z jego prawdziwością w tym stanie (można tu nawiązać do znanego spostrzeżenia greckiego filozofa Parmenidesa – „Byt jest, a niebytu nie ma”).



Rysunek 7. Model Kripke podsystemu interpreter aktywny

Źródło: opracowanie własne.

W celu dokładnego scharakteryzowania predykatów oraz nazw operacji, którymi opieczętowane są stany i przejścia w MKpIA, zestawiono ich semantykę w tabelach 1–2.

Tabela 1. Szczegółowy opis predykatów przypisanych do stanów w MKpIA

Lp.	Formuła	Semantyka	Przykład
1.	$wczytD(X)$	„Wczytano dane, które są X ”, $X \in \{domyślne, poprawne, niepopr\}$	$wczytD(poprawne)$ – wczytano poprawne dane
2.	$przez(X)$	„Prezentowana jest X ”, $X \in \{InterOgolna, InterSzczeg, blad\}$	$przez(InterOgolna)$ – prezentowana jest interpretacja ogólna

Źródło: opracowanie własne.

Należy tutaj zauważyć, że wszystkich możliwych predykatów (zdań), bez ingerencji funktorów zdaniotwórczych, jest dokładnie sześć.

Tabela 2. Szczegółowy opis przejść w MKpIA

Lp.	Nazwa przejścia	Semantyka	Przykład
1.	<i>wprowD(X)</i>	„Wprowadź dane, które są X”, $X \in \{domyslne, poprawne, niepopr\}$	<i>wprowD(X)</i> – wprowadź dane niepoprawne
2.	<i>zamknij(kom)</i>	„Zamknij okno z komunikatem”	<i>zamknij(kom)</i> – zamknij okno z komunikatem, który pojawił się na ekranie
3.	<i>wnioskuj</i>	„Uruchom proces wnioskowania”	<i>wnioskuj</i> – uruchom proces wnioskowania dla wprowadzonych danych

Źródło: opracowanie własne.

Następnie autorzy sporządzili zbiór formuł CTL zgodnych z wymaganiami systemu KBS, dotyczącymi newralgicznych właściwości modelowanego podsystemu interpreter aktywny. W formalnej specyfikacji uwzględniono właściwości takie jak bezpieczeństwo, żywotność i uczciwość (ang. *fairness*). Do weryfikacji wykorzystano klasyczny algorytm *global model checking* (GMC) dla CTL¹⁹, który na wejściu otrzymuje formułę CTL, a w wyniku zwraca zbiór stanów MKpIA, w którym weryfikowana formuła jest spełniona. Jeżeli zbiór ten zawiera wszystkie stany, oznacza to, że dana formuła jest prawdziwa we wszystkich stanach MKpIA. Algorytm należy wykonać dla każdej formuły osobno. Złożoność algorytmu weryfikacji GMC jest wielomianowa i może być on przeprowadzony w czasie (jednostka czasu tutaj jest rozumiana jako operacja jednostkowa) $O(m * n)$, gdzie m to liczba tranzycji w MKpIA, natomiast n jest liczbą podformuł weryfikowanej formuły. Opracowany MKpIA posiada $m = 15$ tranzycji, natomiast wartość n jest generowana osobno dla każdej weryfikowanej formuły. Formuły poddawane procesowi weryfikacji zestawiono w tabeli 3.

Tabela 3. Formalna specyfikacja wymagań systemu KBS w postaci formuł CTL

Numer	Symboliczne wyrażenie	Techniczne znaczenie	Znaczenie potoczne
1.	$M, q_0 \models wczytD(domyslne)$	w stanie q_0 prawdą jest, że wczytane są dane domyślne	interpreter po pierwszym włączeniu ma zawsze wartości domyślne
2.	$\forall_{\phi} F\Phi$	dla każdej formuły prostej Φ prawdą jest, że z dowolnego stanu systemu możemy kiedyś przejść do takiego stanu, w którym formuła ta jest spełniona	system nie blokuje swojego działania (żywotność)

¹⁹ M. Huth, M. Ryan, *Logic in Computer Science: Modelling and Reasoning about Systems*, Cambridge University Press, New York 2004, s. 207–216.

Numer	Symboliczne wyrażenie	Techniczne znaczenie	Znaczenie potoczne
3.	$AG[wczytD(poprawne) \rightarrow EXprez(InterOgólne)]$	w każdym przypadku działania aplikacji od tej pory, jeżeli wprowadzono poprawne dane, to zawsze istnieje możliwość, że w chwili następnej zostanie uruchomiony proces wnioskowania i prezentacja interpretacji ogólnej	wprowadzenie poprawnych danych do systemu gwarantuje możliwość uruchomienia w następnym kroku procesu wnioskowania i prezentacji interpretacji ogólnej (bezpieczeństwo)
4.	$AG[wczytD(niepopr) \rightarrow \sim EXprez(InterOgólne)]$	w każdym przypadku działania systemu od tej pory, jeżeli wprowadzono niepoprawne dane, to nieprawda, że istnieje możliwość w następnej chwili uruchomienia procesu wnioskowania	dla danych niepoprawnych nie można uruchomić procesu wnioskowania (bezpieczeństwo)
5.	$AG[wczytD(niepopr) \rightarrow EXwczytD(poprawne)]$	w każdym przypadku od tej pory, jeżeli dane są niepoprawne, to istnieje możliwość, że w chwili następnej będą poprawne	wprowadzone dane niepoprawne można poprawić w systemie (bezpieczeństwo)
6	$AG[wczytD(poprawne) \rightarrow F(prez(InterOgólne) \vee prez(InterSzczeg))]$	w każdym przypadku od tej pory, jeżeli dane są poprawne, to kiedyś w przyszłości prezentowana będzie interpretacja ogólna lub interpretacja szczegółowa	po wprowadzeniu danych poprawnych zawsze jest pewność, że uruchomimy proces wnioskowania i możemy wyświetlić interpretację ogólną tych danych lub interpretację szczegółową (uczciwość)

Źródło: opracowanie własne.

Każda z formuł ma przyporządkowany unikalny numer. Wyrażenie symboliczne jest przedstawione w formie CTL, natomiast techniczne znaczenie jest rozumiane jako dosłowne znaczenie formuły.

Zestawienie w tabeli 4 pokazuje, że w wyniku działania algorytmu GMC weryfikowane formuły były prawdziwe (kolumna „Rezultat”) we wszystkich stanach, z wyjątkiem formuły nr 1, gdzie prawdziwość wymagano tylko dla stanu q_0 . Oznacza to, że fakty przedstawione w tabeli 3 są prawdziwe w MKpIA, a co za tym idzie w systemie KBS. Udowodniono tym samym, że system nie zawiera błędów z dokładnością do różnicy pomiędzy systemem a jego modelem. Dzięki temu można stwierdzić, iż jakość rozwiązania jest akceptowalna.

Tabela 4. Zestawienie parametrów weryfikacji wymagań formalnych

Numer formuły	n	Czas weryfikacji [$m * n$]	Rezultat
1.	1	1 ²⁰	prawdziwa ²¹ w q_0
2.	2	180 ²²	prawdziwa
3.	7	105	prawdziwa
4.	8	120	prawdziwa
5.	7	105	prawdziwa
6.	8	120	prawdziwa
7.	8	120	prawdziwa

Źródło: opracowanie własne.

6. Podsumowanie

Celem pracy było opracowanie algorytmu systemu opartego na wiedzy wspomagającego decyzje terapeuty przez szybką interpretację wyników z urządzenia EEG wraz z jego implementacją. Cel ten został osiągnięty, jakość aplikacji jest akceptowalna, a uzyskane wyniki istotnie wpływają na wyższą jakość i niższy koszt wypracowanych decyzji terapeuty.

W pracy przedstawiono uwarunkowania środowiskowe opracowywanej aplikacji, wybrane algorytmy, opracowane oprogramowanie oraz ocenę jakości aplikacji metodą *model checking*. Do oceny jakości tą metodą przygotowano schemat weryfikacji KBS, MKpIA oraz zbiór formuł do weryfikacji. Opracowana formalna specyfikacja wymagań systemu KBS oraz zestawienie parametrów weryfikacji wymagań formalnych jednoznacznie wskazują na pozytywną weryfikację zgodności modelu KBS z jego założeniami.

W dalszych pracach autorzy planują wprowadzenie interpretacji również z uwzględnieniem analizy widmowej sygnałów z EEG. Trwają prace nad dołączeniem do zbioru danych wejściowych sygnałów pochodzących z innych narządów człowieka, np. serca czy gałek ocznych. Ponadto, kluczowe będzie rozwinięcie elementu sieciowości prowadzonych interpretacji oraz rozbudowanie bazy wiedzy.

²⁰ Weryfikacja polega jedynie na sprawdzeniu, czy w stanie q_0 formuła *wczytD(domyślne)* jest prawdziwa.

²¹ Rezultat „prawdziwa” oznacza, że dana formuła jest prawdziwa we wszystkich stanach MKpIA.

²² Czas weryfikacji dla tej formuły jest dodatkowo pomnożony przez liczbę wszystkich formuł prostych modelu MKpIA, która jest równa 6 (zgodnie z tabelą 1).

Bibliografia

- Błasiak W., Godlewska M., Rosiek R., Wcisło D., *Nowe technologie w badaniach edukacyjnych*, w: *Człowiek, media, edukacja*, red. J. Morbitzera, E. Musiał, Katedra Technologii i Mediów Edukacyjnych, Uniwersytet Pedagogiczny, Kraków 2013.
- Clarke E.M., *Model Checking*, w: *Handbook of Automated Reasoning*, red. A. Robinson, A. Voronkov, Elsevier BV, Amsterdam 2001, s. 1635–1790.
- Clarke E.M., *The Birth of Model Checking*, w: *25 Years of Model Checking*, red. O. Grumberg, H. Veith, „Lecture Notes in Computer Science”, cz. 5000, Springer, Berlin–Heidelberg 2008, s. 1–26.
- Emerson E.A., Clarke E.M., *Using branching time logic to synthesize synchronization skeletons*, „Science of Computer Programming” 1982, vol. 2, s. 241–266.
- Filipkowski P., Horodelski M., *Zastosowanie rozproszonej platformy modelowania i symulacji do analizy informacji medycznych na bieżąco*, „Roczniki” Kolegium Analiz Ekonomicznych, z. 35, Oficyna Wydawnicza SGH, Warszawa 2014.
- Filipkowski P., Horodelski M., Kozłowski A., Kurowski P., Lasyk J., Mruk W., Oleszczuk D., *Technologie społeczeństwa informacyjnego*, cz. 1, „Przegląd Techniczny” 2012, nr 22.
- Filipkowski P., Horodelski M., Kozłowski A., Kurowski P., Lasyk J., Mruk W., Oleszczuk D., *Technologie społeczeństwa informacyjnego*, cz. 2, „Przegląd Techniczny” 2012, nr 23–24.
- Filipkowski P., Kasperek M., Nowakowski J., *System operacyjny Linux tsibuntu live*, w: *LabTSI™ – platforma modelowania i symulacji*, red. A. Janicki, Wydawnictwo KUL, Lublin 2011.
- Filipkowski P., Kasperek M., Nowakowski J., *Wirtualna Maszyna Równoległa*, w: *LabTSI™ – platforma modelowania i symulacji*, red. A. Janicki, Wydawnictwo KUL, Lublin 2011.
- Horodelski M., *Narzędzie modelowania i symulacji Scilab*, w: *LabTSI™ – platforma modelowania i symulacji*, red. A. Janicki, Wydawnictwo KUL, Lublin 2011.
- Huth M., Ryan M., *Logic in Computer Science: Modelling and Reasoning about Systems*, Cambridge University Press, New York 2004.
- Inteligentne systemy w zarządzaniu – teoria i praktyka*, red. J.S. Zieliński, Wydawnictwo Naukowe PWN, Warszawa 2000.
- Jaśkowski P., *Zarys psychofizjologii*, Wyższa Szkoła Finansów i Zarządzania w Warszawie, Warszawa 2004.
- Kripke S., *Semantical Considerations on Modal Logic*, „Acta Philosophica Fennica” 1963, vol. 16, s. 83–94.
- Mulawka J.J., *Systemy ekspertowe*, WNT, Warszawa 1996.
- Wrycza S., Marcinkowski B., Wyrzykowski K., *Język UML 2.0 w modelowaniu systemów informatycznych*, Helion, Gliwice 2006.

* * *

Application of knowledge-based system for fast identification of EEG frequency

Summary

In the paper, the authors solve the problem of fast interpretation of data from selected psychophysiological devices, dedicated to research on semi-trained respondents acting in stress environment, making highly responsible decisions in short time. Results from EEG (electroencephalography), among others, are evaluated by therapists according to their knowledge. Classic methods are acceptable for accidental cases, but in group cases they prolong the time of the analysis. In the paper the authors present a method of digital support of therapist's or trainer's decisions in the interpretation of data received from medical devices. The authors propose an algorithm and its implementations for visualizations of given interpretations, which help to increase the speed of biofeedback for therapist. Additionally, the authors propose a built-in platform which is mobile, scalable and allows using engineering tools, including computational intelligence in future work.

Method: An environment of Knowledge-Base System 2014 for the automatic interpretation of data from sensors has been developed together with a batch system with an internal knowledge base, ready to work in LiveCD mode in accordance with the IEEE829 standard, providing not only mobility, but also the acceleration of various applications integrated with psychophysiological devices.

Quality evaluation: To examine the compatibility of KBS system with the requirements, an abstract verification based on the model checking method has been used.

Keywords: decision support systems, knowledge-based system, psychophysiological device, EEG, scilab gui, batch operating systems, model checking

