

ROZPROSZONY SYSTEM WIELOAGENTOWY WSPOMAGAJĄCY PROJEKTOWANIE WYBRANYCH UKŁADÓW STEROWANIA

Przemysław SPYCHALSKI¹, Ryszard ARENDT²

Politechnika Gdańska, Wydział Elektrotechniki i Automatyki

1. tel.: 58 347 2139, e-mail: przspsych@student.pg.gda.pl

2. tel.: 58 347 2157, e-mail: ryszard.arendt@pg.gda.pl

Streszczenie: W artykule przedstawiono system wieloagentowy wspomagający projektowanie wybranych układów sterowania sterami strumieniowymi. Zaimplementowany system składa się z czterech niezależnych programów komputerowych (agentów), które poprzez realizację unikalnych algorytmów oraz wzajemną komunikację dobierają odpowiednią strukturę i elementy układu sterowania oraz po przeprowadzeniu badań symulacyjnych dokonują oceny jakości otrzymanego rozwiązania projektowego.

Słowa kluczowe: system wieloagentowy, wspomaganie projektowania, stery strumieniowe, obliczenia symulacyjne.

1. WPROWADZENIE

Koncepcja oraz pierwsze implementacje systemów wieloagentowych pojawiły się już pod koniec lat osiemdziesiątych XX wieku, jednak podejście to jest nadal aktualne i rozwijane w środowiskach akademickich, jak również w zastosowaniach komercyjnych (przykładem może być oprogramowanie antywirusowe bazujące na architekturze agentowej, aplikacje wspomagające kontrolę ruchu lotniczego oraz zarządzanie przedsiębiorstwem) [1, 2].

Systemy wieloagentowe stosowane są wszędzie tam, gdzie zachodzi potrzeba rozwiązania problemu o naturze rozproszonej lub złożonej obliczeniowo. Programowanie agentowe cechuje poziom abstrakcji wyższy, niż np. w programowaniu obiektowym i strukturalnym. Przewagą takiego podejścia w stosunku do innych paradygmatów programowania jest zdecentralizowane przeprowadzanie obliczeń, dzięki czemu możliwe jest równoległe wykonywanie poszczególnych części algorytmów (również na odrębnych jednostkach sprzętowych) i znacznie szybsze opracowanie rozwiązania przez system.

Kolejnym aspektem przemawiającym na korzyść systemów wieloagentowych jest odporność na zakłócenia oraz błędy. Obliczenia wykonywane są zazwyczaj redundantnie, po czym dokonywana jest ocena jakości otrzymanych rozwiązań. Bazując na określonym kryterium wybrany zostaje najlepszy z dostępnych rezultatów. Pozwala to uniknąć błędów, które mogą wystąpić w programowaniu obiektowym, przez to że każdy obiekt zakłada poprawność danych otrzymanych od innego obiektu. Dane w systemie wieloagentowym przesyłane są w sposób asynchroniczny i nadmiarowy. Uszkodzenie pewnego fragmentu nie skutkuje zanikiem funkcjonalności systemu jako całości, co jest bardzo istotnie np. w środowisku sieciowym [1, 2].

Podstawową jednostką w systemie jest agent (ang. *Software Agent*). Nie istnieje spójna, jednoznaczna definicja agenta, jednak można stwierdzić, iż jest to autonomiczny program komputerowy, posiadający zdolność wykonywania określonego zakresu działań oraz komunikowania się z innymi agentami, w celu wspólnego rozwiązania danego problemu. Najczęściej pojedynczy agent nie posiada dostatecznej wiedzy oraz umiejętności do samodzielnego rozwiązania postawionego zadania. Dodatkowymi cechami agentów, często wymienianymi w literaturze są [1, 2]:

- zdolność uczenia się,
- zdolność do zachowań społecznych,
- zdolność adaptacji do nowych warunków,
- zdolność do reagowania w czasie rzeczywistym,
- zdolność posługiwania się językiem naturalnym.

W ramach jednego systemu może działać od kilku do nawet kilku tysięcy agentów. Do zarządzania agentami wchodzącymi w skład systemu służy platforma wieloagentowa. Jest to oprogramowanie stosowane do podglądu interakcji oraz komunikacji pomiędzy agentami, dodawania nowych instancji agentów do systemu, zawieszania działania agentów lub w razie konieczności usuwania agentów z systemu. Standardy działania platform (jak również całych systemów wieloagentowych) określa organizacja FIPA (ang. *Foundation for Intelligent Physical Agents*). Do głównych wymagań stawianych platformie wieloagentowej można zaliczyć [1, 2]:

- stabilność (ang. *Stability*),
- krzepkość (ang. *Robustness*),
- użyteczność (ang. *Usability*),
- wydajność (ang. *Performance*),
- skalowalność (ang. *Scalability*),
- bezpieczeństwo (ang. *Security*),
- zgodność ze standardami (ang. *Compatibility*).

Opisane założenia stanowiły bazę do implementacji systemu wieloagentowego wspomagającego projektowanie. Projektowanie to proces złożony, wymagający dużej mocy obliczeniowych oraz integracji wielu narzędzi, dlatego rozproszona architektura systemu stanowi tutaj dużą zaletę. Utworzony system został wyspecjalizowany w zakresie wspomaganie projektowania układów sterowania sterów strumieniowych statków. Umożliwia on dobór odpowiedniej struktury (topologii) i elementów składowych układu oraz udostępnia algorytmy określające jakość uzyskanego rozwiązania projektowego.

2. UKŁADY STEROWANIA STERAMI STRUMIENIOWYMI

Ster strumieniowy jest to urządzenie mechaniczne znajdujące się na dziobie i/lub rufie statku, którego zadaniem jest zwiększenie możliwości manewrowych jednostek pływających. Poprzez wytworzenie siły naporu prostopadle do osi okrętu możliwe jest bardziej precyzyjne manewrowanie w portach przy małych prędkościach, dosuwanie jednostki do nabrzeża, a także sprawniejsze wykonywanie nawrotów. Najczęściej stery strumieniowe mają postać tunelu umiejscowionego poniżej poziomu wody z zamontowaną wewnątrz śrubą napędową.

Podczas rozruchu sterów strumieniowych pobierana jest moc elektryczna, która może wpłynąć niekorzystnie na pracę elektrowni okrętowej. Aby tego uniknąć stosowane są układy sterowania regulujące charakterystyki prądowe podczas załączenia i pracy sterów. W procesie projektowania uwzględnia się pobór mocy przez elementy składowe oraz dobiera odpowiednią strukturę układu w zależności od sposobu rozruchu steru strumieniowego.

Struktura z rozruchem autotransformatorowym zapewnia dynamiczne sterowanie momentem poprzez zwiększenie liczby odczepów autotransformatora. Przy rozruchu autotransformatorowym występuje dwukrotne przekroczenie prądu nominalnego silnika indukcyjnego, śruba ustawiona jest w pozycji jałowej. Elementy struktury: okrętowy silnik spalinowy, generator, silnik indukcyjny, wał oraz śruba o skoku nastawnym.

Struktura z rozruchem bezpośrednim jest najprostszą strukturą stosowaną do rozruchu tunelowych sterów strumieniowych, charakteryzującą się wysokim momentem rozruchowym. Występuje tutaj znaczne (sześciokrotne) przekroczenie prądu nominalnego silnika indukcyjnego oraz podobnie jak w przypadku rozruchu autotransformatorowego śruba ustawiona jest w pozycji jałowej. Elementy struktury: okrętowy silnik spalinowy, generator, silnik indukcyjny, wał oraz śruba o skoku nastawnym.

Struktura z rozruchem gwiazda-trójkąt charakteryzuje się znacznym obniżeniem momentu rozruchowego, który jest w przybliżeniu trzykrotnie niższy, niż w przypadku rozruchu bezpośredniego. Zachodzi też proporcjonalnie niższe (trzykrotne) przekroczenie prądu nominalnego silnika indukcyjnego podczas rozruchu. Elementy struktury: okrętowy silnik spalinowy, generator, przełącznik gwiazda-trójkąt realizowany przez dwa styczniki, silnik indukcyjny, wał oraz śruba o skoku nastawnym.

Struktura z rozruchem soft-start zapewnia płynną regulację prądu podczas rozruchu steru strumieniowego poprzez zastosowanie specjalnych modułów elektronicznych (soft-starterów). Dzięki temu występuje najniższe przekroczenie wartości prądu spośród wszystkich struktur, jednak czas rozruchu ulega wydłużeniu. Elementy struktury: okrętowy silnik spalinowy, generator, soft-starter realizowany np. przez układ tyrystorów, silnik indukcyjny, wał oraz śruba o skoku nastawnym.

Wybór odpowiedniej struktury układu sterowania rozruchem sterów strumieniowych oraz jego elementów składowych nie zapewnia jednak otrzymania optymalnych charakterystyk dynamicznych układu. Zaimplementowany system wieloagentowy oprócz doboru wcześniej wspomnianych komponentów daje możliwość przeprowadzenia badań symulacyjnych utworzonej struktury i na tej podstawie zbadanie właściwości dynamicznych układu w warunkach laboratoryjnych.

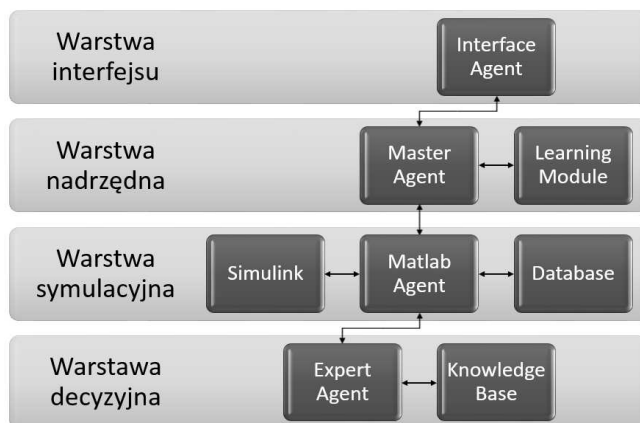
3. SYSTEM WIELOAGENTOWY WSPOMAGAJĄCY PROJEKTOWANIE

Znane są implementacje systemów służących do częściowej automatyzacji procesu projektowania układów automatyki statków. Niektóre z nich ograniczają się jedynie do utworzenia narzędzia, które ułatwia wymianę informacji i dzielenie się wiedzą przez inżynierów projektantów [3]. Zaimplementowany został również system ekspertowy, który rozwiązuje problem automatycznego doboru struktury i elementów układu sterowania sterami strumieniowymi [4, 5]. Składa się on z interfejsu użytkownika, bazy wiedzy, bazy danych elementów oraz aplikacji symulacyjnej.

Wspomniany system ekspertowy charakteryzuje się jednak scentralizowaną strukturą. Jest to samodzielna aplikacja (ang. *Standalone application*) przez co posiada pewne wady oraz ograniczenia z tego wynikające:

- Centralizacja algorytmów w obrębie jednostki sprzętowej prowadzi do jej nadmiernego obciążenia podczas obliczeń symulacyjnych;
- Z systemu może korzystać naraz tylko jeden użytkownik, aktualizacja bazy danych okresowo blokuje możliwość pracy z systemem ekspertowym;
- Jednowątkowość aplikacji, która uniemożliwia zrównoleglenie obliczeń, przez co poszczególne etapy projektowe nie są realizowane współbieżnie;
- Brak modułowości prowadzi do trudności w lokalizacji błędów i komplikuje dodawanie nowych funkcjonalności do systemu.

Podjęcie wieloagentowe pozwala uniknąć wcześniej wspomnianych ograniczeń. Główną zaletą systemów wieloagentowych jest możliwość uruchamiania oraz równoległego działania niezależnych instancji – agentów. System składa się z czterech podstawowych typów agentów: agenta interfejsu (Interface Agent), agenta nadrzędnego (Master Agent), agenta symulacyjnego (Matlab Agent) oraz agenta decyzyjnego (Expert Agent). Struktura utworzonego systemu przedstawiona została na rysunku 1 [6]:



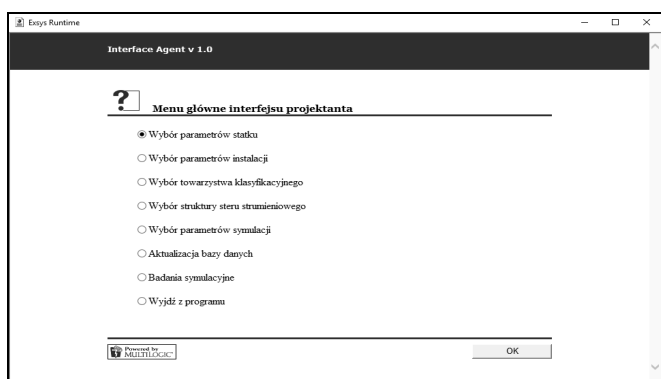
Rys. 1. System wieloagentowy wspomagający projektowanie [6]

System wieloagentowy ma architekturę warstwową. Kiedy czas przetwarzania danych przez daną warstwę jest zbyt długi, wówczas możliwe jest uruchomienie kolejnej instancji agenta na innym komputerze. System posiada mechanizm kolejkowania, dzięki któremu zapytanie od wyższej warstwy przetwarzane jest przez najmniej obciążoną jednostkę warstwy niższej. W dalszej części artykułu przedstawiono charakterystykę poszczególnych typów agentów oraz przykładowy proces projektowania.

3.1. Interface Agent

Agent interfejsu ma za zadanie zbieranie danych oraz umożliwienie interakcji użytkowników z systemem wieloagentowym. Dodatkowo odpowiada on za wstępną walidację wprowadzonych danych. Uruchomienie kilku instancji agentów tego typu pozwala na równoczesną pracę kilku inżynierów projektantów z systemem, przy czym standardowo na każdym komputerze działa tylko jeden Interface Agent. Komunikacja polega na zadawaniu pytań użytkownikowi po wybraniu odpowiedniej kategorii projektowej. Dzięki temu formułowane są założenia co do projektowanego układu sterowania.

Użytkownik ma możliwość wprowadzenia parametrów dotyczących projektowanego statku, cech jego instalacji elektrycznej i ograniczeń wynikających z wymagań towarzystwa klasyfikacyjnego. Możliwa jest także aktualizacja bazy danych elementów składowych z poziomu interfejsu, a także wybranie przykładowej struktury układu sterowania sterami strumieniowymi i przeprowadzenie dla niej badań symulacyjnych z odpowiednimi parametrami. Dane zapisywane są w postaci tabelarycznej i po zebraniu wszystkich niezbędnych informacji wysyłane są one dalej do agenta nadrzędnego. Okno główne agenta interfejsu użytkownika systemu przedstawiono na rysunku 2 [5, 6]:



Rys. 2. Okno główne agenta interfejsu użytkownika [5]

3.2. Master Agent

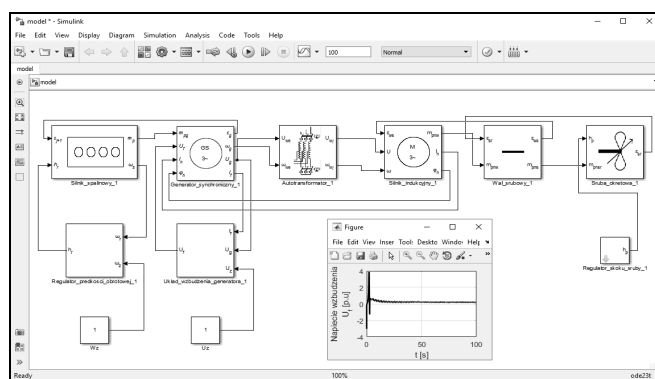
Agent nadrzędny (master) jest odpowiedzialny za nadzorowanie procesów projektowych. Posiada on także algorytm uczący, który w niektórych przypadkach znacząco skraca czas opracowania rozwiązania przez system wieloagentowy. Agent master otrzymując dane wejściowe od poszczególnych agentów interfejsu kataloguje je w pliku w postaci tablicy mieszającej (ang. *Hash table*). Następnie, po otrzymaniu raportu z analizy utworzonego układu sterowania sterami strumieniowymi (etap realizowany przez agenta symulacyjnego i agenta ekspertowego), dokonuje jego asocjowania z odpowiednimi danymi wejściowymi.

Proces uczenia na podstawie danych wejściowych służy do tego, że jeżeli w przyszłości użytkownik wprowadzi dane wejściowe, a algorytm odnajdzie taki sam wzorzec w pliku, to wówczas ponownie przeprowadzanie procesu badań symulacyjnych i oceny przydatności rozwiązania nie jest konieczne, ponieważ początkowe założenia projektowe są takie same. Pozwala to znacząco skrócić proces projektowania, w zasadzie użytkownik natychmiast otrzymuje gotowy raport z analizy danej struktury sterowania bazujący na danych historycznych. Jeżeli jednak dane są unikalne (wzorzec nie został rozpoznany) to następuje przekierowanie tych informacji do agenta symulacyjnego (forma tabelaryczna zostaje zachowana).

3.3. Matlab Agent

Agent symulacyjny bazuje na oprogramowaniu Matlab Simulink. Odpowiada on za badania symulacyjne utworzonych struktur układów sterowania. Po otrzymaniu danych wejściowych, zawierających wytyczne do projektowanej struktury, następuje połączenie z bazą danych elementów składowych, z której importowane są niezbędne komponenty (np. model generatora lub silnika indukcyjnego). Parametry wczytanych elementów układu sterowania zaczerpnięte zostały z dokumentacji technicznej producentów lub (w przypadku braku niezbędnych informacji) dobrane przez algorytm ewolucyjny.

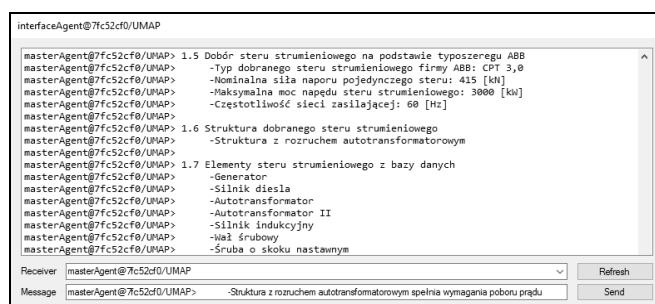
W dalszej kolejności następuje zestawienie połączeń pomiędzy blokami modelu i uruchomienie symulacji. Wyniki badań symulacyjnych dostępne są w postaci charakterystyk przedstawiających np. spadek napięcia na generatorze podczas rozruchu steru strumieniowego, moment obrotowy, nastawę listwy paliwowej i wiele innych. Wyniki te przesyłane są dalej do agenta decyzyjnego w celu weryfikacji optymalności otrzymanego rozwiązania projektowego. Przykładowy układ sterowania zamodelowany przez agenta symulacyjnego na podstawie wytycznych wprowadzonych przez użytkownika oraz części wyników symulacji przedstawiono na rysunku 3 [5, 6]:



Rys. 3. Model układu utworzony przez agenta symulacyjnego [5]

3.4. Expert Agent

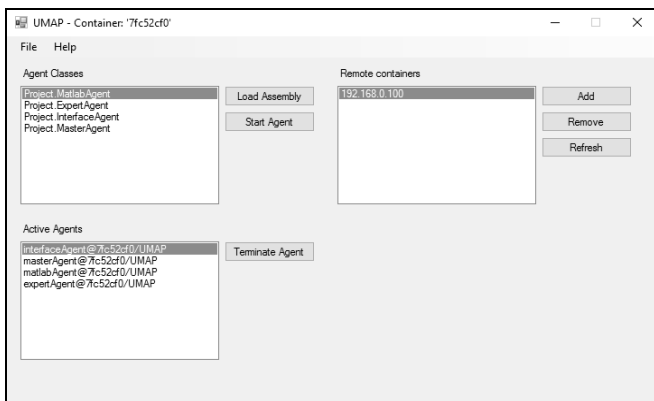
Agent decyzyjny (expert) na podstawie reguł zawartych w bazie wiedzy konfrontuje wyniki badań symulacyjnych z wytycznymi towarzystwa klasyfikacyjnego, którego ocenie podlega projektowany statek. Ograniczenia te dotyczą np. dopuszczalnej wartości spadku napięcia na generatorze podczas rozruchu sterów strumieniowych etc. Agent decyzyjny generuje i wysyła raport z analizy utworzonego układu sterowania sterami strumieniowymi. Raport ten zawiera niezbędne informacje na temat zaprojektowanego systemu, jego struktury, elementów, wyniki badań symulacyjnych oraz wytyczne dla projektanta [6]:



Rys. 4. Raport z analizy układu sterowania sterów strumieniowych

3.5. Universal Multi-Agent Platform (UMAP)

System oparty został na uniwersalnej platformie wieloagentowej UMAP, która daje możliwość uruchomienia (w każdej warstwie zaimplementowanego systemu) nieograniczonej ilości współpracujących ze sobą agentów. Pozwala to na równoczesne prowadzenie wielu procesów konstrukcyjnych. Dzięki temu czas potrzebny do otrzymania rozwiązań kilku zadań projektowych jest tylko nieznacznie wyższy w stosunku do czasu przetworzenia pojedynczego żądania przez wspomniany system scentralizowany. Komunikacja pomiędzy poszczególnymi jednostkami agentowymi zrealizowana została w oparciu o komunikaty w formacie XML oraz obsługę gniazd sieciowych (ang. *Socket*). Platforma UMAP napisana została w języku C# z wykorzystaniem .NET Framework zgodnie ze specyfikacją FIPA. Posiada ona graficzny interfejs użytkownika ułatwiający zarządzanie systemem wieloagentowym [7]:



Rys. 5. Uniwersalna platforma wieloagentowa UMAP [7]

Uniwersalna platforma wieloagentowa UMAP składa się z trzech podstawowych elementów:

- Usługa katalogowania agentów DF (ang. *Directory Facilitator*);
- System zarządzania agentami AMS (ang. *Agent Management System*);
- System transportu komunikatów MTS (ang. *Message Transport System*);

Platforma UMAP tworzy kontenery, wewnątrz których uruchamiane są procesy agentów. Każdy proces składa się domyślnie z kilku wątków (w tym przypadku są to cztery). Jeden wątek jest zarezerwowany do obsługi wiadomości ACL (ang. *Agent Communication Language*), natomiast pozostałe realizują współbieżnie zdefiniowane algorytmy projektowe, co przekłada się na wzrost wydajności na poziomie 172% w stosunku do aplikacji jednowątkowej przy takiej samej mocy obliczeniowej (Intel Core i7 vPro).

4. PODSUMOWANIE

Systemy wieloagentowe znalazły zastosowanie przy rozwiązywaniu problemów o naturze rozproszonej lub złożonych obliczeniowo. Jednym z takich zagadnień jest projektowanie układów sterowania sterami strumieniowymi, wśród których można wyróżnić m.in. struktury z rozruchem bezpośrednim, autotransformatorowym, gwiazda-trójkąt oraz soft-start. Rezultatem działania zaimplementowanego systemu wieloagentowego jest dobór odpowiedniej struktury układu sterowania, jego elementów składowych, przeprowadzenie badań symulacyjnych wraz z optymalizacją parametrów oraz ocena przydatności otrzymanego rozwiązania projektowego. Dzięki temu system stanowi funkcjonalne narzędzie, które jest gotowe do zastosowania na przykład w biurach projektowych i centrach badawczych.

Dzięki rozproszonej architekturze zaimplementowany system wieloagentowy umożliwia prowadzenie kilku procesów projektowych jednocześnie. Zaletą takiego podejścia jest skrócenie czasu otrzymania gotowych rozwiązań projektowych (w porównaniu z aplikacją scentralizowaną) oraz możliwość pracy kilku użytkowników w systemie. Poszczególne zadania projektowe realizowane są przez cztery podstawowe typy agentów: agenta interfejsu (Interface), agenta nadrzędnego (Master), agenta symulacyjnego (Matlab) i agenta decyzyjnego (Expert). Instancje agentów są wielowątkowe, przez co osiągnięta zostaje wydajność przetwarzania zapytań większa o 172% w stosunku do scentralizowanej aplikacji jednowątkowej.

5. BIBLIOGRAFIA

1. Weiss G.: Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, The MIT Press 1999
2. Wooldridge M.: An Introduction to Multiagent Systems Second Edition, John Wiley & Sons 2002
3. Xiangzhong F.: Ship Collaborative Design Based on Multi-agent and Ontology, Springer 2008
4. Arendt R., van Uden E.: A decision-making module for aiding ship system automation design: A knowledge-based approach, Expert Systems with Applications 2011
5. Arendt R., Kopczyński A.: Using an expert system for a chosen ship power subsystems design, Przemysłowy Instytut Automatyki i Pomiarów PIAP 2007
6. Arendt R., Sychalski P.: An application of multi-agent system for ship's power systems design, Proceedings of 20th International Conference Transport Means 2016
7. Mrozek D., Małysiak-Mrozek B., Waligóra I.: UMAP - A Universal Multi-Agent Platform for .NET Developers, Springer 2014

DISTRIBUTED MULTI-AGENT SYSTEM FOR AIDED DESIGN OF CHOSEN CONTROL SYSTEMS

In this paper the distributed multi-agent system for aided design of chosen control systems is presented. Implemented system consists of four independent software agents: Interface Agent (gathers and initially validates the data from user designer), Master Agent (supervises the design process by delegating tasks to particular agents and enables deployment of machine learning algorithms), Matlab Agent (loads the necessary items from the database, creates a control system model and conducts dynamic calculations for the designed structure), Expert Agent (evaluates the simulation results regarding the suitability of generated solution and provides a report for system user). The outcome of the system activity is a ready-made solution concerning an appropriate structure and elements of the thruster control system.

Keywords: multi-agent system, aided design, ship thrusters, simulations.