

Agentowy system wielomodelowy do zarządzania grupą robotów mobilnych

Autoreferat rozprawy doktorskiej

Autor: mgr inż. Wojciech Turek

Promotor: dr hab. inż. Krzysztof Cetnarowicz, prof. n. AGH

1. Wstęp

W ciągu ostatnich dziesięcioleci nastąpił bardzo znaczący rozwój technologii wytwarzania urządzeń mechanicznych, które są w stanie przemieszczać się, postrzegać otaczający środowisko i wykonywać w nim różnego rodzaju czynności. Urządzenia tego typu, zwane robotami mobilnymi, mogą mieć bardzo różnorodne przeznaczenie, które determinuje rodzaj stosowanej konstrukcji, rozmiary, rodzaj napędu czy stosowanych sensorów i efektorów.

Sz szczególnie duże nadzieje wiąże się z zastosowaniami grup robotów autonomicznych, które będą w stanie samodzielnie wykonywać złożone zadania. Przez autonomię robota rozumie się jego niezależność od elementów zewnętrznych, nie będących częścią robota. Robot może być autonomiczny pod względem źródła zasilania, jednostki sterującej, percepcji otoczenia i możliwości wpływania na otoczenie. Teoretyczne możliwości zastosowań grup współpracujących, autonomicznych robotów mobilnych wydają się niemal nieograniczone. Mogą dotyczyć takich obszarów tak odległych jak eksploracja kosmosu, lub tak codziennych jak sprzątanie supermarketu.

Łatwo jednak zauważyć, że pomimo rosnących możliwości sprzętu, roboty mobilne ciągle nie są obecne w codziennym życiu, a ich wykorzystanie w przemyśle jest bardzo ograniczone. Większość praktycznych zastosowań robotów mobilnych ogranicza się do wykorzystywania jednego, nieskomplikowanego urządzenia, bądź nie wykorzystuje możliwości współpracy wielu robotów przy wykonywaniu wspólnego zadania. Ogranicza to znacząco spektrum możliwych zastosowań.

Przyczyną jest bardzo wysoki stopień skomplikowania niezbędnego oprogramowania sterującego oraz **brak metodologii tworzenia złożonych systemów wielorobotowych**. Każdy, nawet najprostszy program sterujący robotami mobilnymi, musi rozwiązywać problemy z wielu dziedzin, takich jak analiza i przetwarzanie sygnałów z sensorów, komunikacja bezprzewodowa, planowanie i wykonanie trasy, budowanie modelu środowiska czy sterowanie różnego typu efektorami. Złożoność każdego z tych zagadnień spowodowała, że aktualnie prowadzone badania nad oprogramowaniem robotów skupiają się nad rozwiązywaniem odseparowanych pod-problemów. Istnieje sporo prac poświęconych zagadnieniom nawigacji reaktywnej, budowania mapy czy koordynacji ruchu, które opisują skuteczne algorytmy rozwiązywania poszczególnych zagadnień.

Wdrożenie praktycznych zastosowań robotów mobilnych wymaga jednak, by rozwiązania poszczególnych problemów zostały zintegrowane w system. W niniejszej pracy system złożony z pewnej liczby robotów oraz sterującego nimi oprogramowania będzie postrzegany jako system informatyczny wykorzystujący roboty do wykonywania zadań w świecie rzeczywistym. Każdy system informatyczny musi spełnić szereg wymagań. Podstawowe wymagania opisują stawiane przed systemem zadania, czyli czynności, jakie system ma wykonywać. Są one nazywane wymaganiami funkcjonalnymi. Jednak poza zestawem posiadanych funkcji, system musi spełniać również szereg wymagań pozafunkcyjnych, które nie są bezpośrednio związane z celem jego istnienia, ale mają zagwarantować wysoką jakość systemu.

Do najważniejszych wymagań pozafunkcyjnych, jakie powinien spełniać system informatyczny wykorzystujący roboty mobilne, można zaliczyć:

- Skalowalność, czyli możliwość powiększania systemu o dodatkowe elementy znanego typu. Przykładem powiększania skali systemu może być zwiększanie liczby robotów czy poszerzanie środowiska ich działania.
- Rozszerzalność, czyli łatwość dodawania nowych funkcji systemu. Rozszerzeniem systemu może być na przykład dodanie nowych rodzajów zadań, które roboty są w stanie wykonywać.

- Wysoka dostępność i odporność na awarie, która gwarantuje nieprzerwane działanie systemu również w sytuacjach wyjątkowych. System powinien być w stanie samodzielnie wykryć i poprawnie zareagować na sytuacje takie jak awaria robota czy błąd komponentu programowego.
- Łatwość ponownego wykorzystywania elementów, która pozwala na szybkie tworzenie nowych rodzajów systemów. Wysoki stopień skomplikowania oprogramowania niezbędnego do sterowania robotami powoduje, że konieczne jest korzystanie z gotowych komponentów. Zmniejsza to znacznie koszty i skraca czas realizacji systemu.

2. Teza pracy

Celem pracy było wykazanie, że przy zastosowaniu odpowiedniej metodologii projektowania i implementowania, możliwe jest stworzenie systemu zarządzającego grupami robotów, który będzie spełniał określone wymagania funkcjonalne oraz będzie się charakteryzował wymienionymi powyżej cechami pozafunkcjonalnymi. Zaproponowana metodologia jest oparta o zastosowanie paradygmatu agentów programowych. Wykorzystuje ona ideę rozdzielania przestrzeni działania komponentów programowych od przestrzeni działania robotów oraz koncepcję jednoczesnego wykorzystania wielu różnych modeli środowiska działania robotów.

W celu wykazania tak postawionej tezy zrealizowany został prototypowy system nawigacji robotów mobilnych. Został on przetestowany z wykorzystaniem rzeczywistych robotów oraz systemu symulacji robotów. Przeprowadzone testy pozwalają stwierdzić, że zaproponowane w pracy podejście pozwala na tworzenie systemów zarządzających robotami, które cechują się pożądanymi własnościami pozafunkcjonalnymi.

3. Koncepcja projektowania systemów agentowych zarządzających robotami

3.1. Separacja przestrzeni działania robotów i agentów

Oddzielenie abstrakcji agenta od obiektu robota osiągnięte zostało poprzez zdefiniowanie przestrzeni działania agentów, która jest odseparowana od przestrzeni działania robotów. Jest ona nazywana przestrzenią wirtualną, lub cyberprzestrzenią. Cyberprzestrzeń to podstawowe środowisko istnienia i funkcjonowania agentów programowych. Jest realizowane jako maszyna wirtualna łącząca wszystkie komputery dostępne dla agentów systemu. Agenty mogą działać na dowolnym z połączonych komputerów, mogą się ze sobą komunikować za pomocą asynchronicznych wiadomości, mogą korzystać z usług platformy agentowej oraz zewnętrznych interfejsów programowych (np. interfejsów efektorów i sensorów robota).

Separacja komponentu programowego, jakim jest agent, od urządzenia działającego w przestrzeni fizycznej, jakim jest robot, ma charakter logiczny, a nie fizyczny. Platforma agentowa może integrować również komputery lokalne mobilnych robotów, pozwalając agentom na wykonywanie programu na tych komputerach. Możliwe jest zrealizowanie prezentowanego podejścia z wykorzystaniem wyłącznie komputerów robotów, tworząc w ten sposób system wielorobotowy, który nie wymaga infrastruktury w środowisku działania.

Każdy z robotów w systemie musi definiować interfejs programowy udostępniający wszystkie operacje wspierane przez sprzęt. Interfejsy różnych typów robotów będą różne, ale wszystkie muszą być dostępne dla przynajmniej jednego agenta w platformie agentowej. Agent ten implementuje zestaw usług robota udostępnionych innym agentom w systemie. Pozostałe agenty mogą traktować te usługi sprzętowe podobnie jak usługi czysto programowe – nie muszą być świadome sposobu ich realizacji. Dzięki takiemu podejściu proaktywna logika systemu, która jest odpowiedzialna za realizację funkcjonalności, może być niezależna od rodzaju robotów wykonujących czynności w rzeczywistym środowisku.

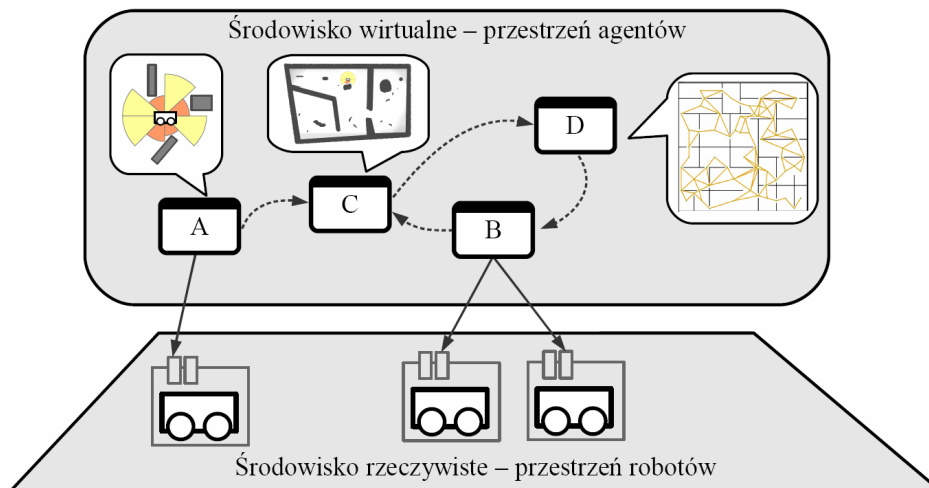
Należy zaznaczyć, że agent wykorzystujący interfejs pojedynczego robota jest częścią systemu agentowego, a nie robota. Różnica jest bardzo znacząca, ponieważ umożliwia wykorzystanie cennych zalet, jakie oferuje paradygmat agentowy. W szczególności pozwala na:

- Wykonywanie programu agenta na dowolnym komputerze w platformie agentowej oraz przenoszenie programu agenta pomiędzy różnymi komputerami. Pozwala to na udostępnienie algorytmom potrzebnej mocy obliczeniowej i ograniczenie wymagań dotyczących autonomicznych komputerów robotów. Rozbudowa systemu o nowe funkcje, które mogą wymagać złożonych obliczeń będzie się mogła odbyć bez wymiany lub modyfikacji robotów.
- Monitorowanie poprawności działania agenta sterującego robotem przez inne agenty w platformie. Awaria wbudowanego programu sterującego robotem unieruchamia urządzenie i czyni go bezużytecznym z punktu widzenia systemu. Możliwość monitorowania poprawności działania agenta sterującego robotem może zabezpieczyć system przed taką sytuacją, zwiększając jego niezawodność.
- Wykrywanie uszkodzeń sprzętu przez agenta sterującego wadliwym robotem. Agent uruchomiony na zewnętrznym komputerze jest w stanie wykryć uszkodzenie robota i uruchomić odpowiednie procedury obsługi błędu.
- Modyfikowanie sposobu działania robota bez konieczności usuwania go z systemu. Podmiana agenta sterującego robotem może zmienić sposób wykonywania czynności przez robota oraz dodać nowe usługi robota. Platforma agentowa umożliwia dokonanie zmiany programu agenta bez konieczności zatrzymywania systemu.

Wprowadzenie logicznej separacji pomiędzy fizycznym urządzeniem, jakim jest robot, a komponentem programowym, który robotem steruje, jest więc podstawowym warunkiem osiągnięcia pożądanych własności pozafunkcjonalnych systemów zarządzających działaniem robotów.

3.2. Jednoczesne wykorzystanie różnych modeli

Zastosowanie paradygmatu agentów programowych wprowadza modularyzację systemu. Funkcjonalność jest podzielona pomiędzy poszczególne agenty, a asynchroniczna komunikacja wymusza obsługiwane sytuacji, w której jeden z agentów nie działa poprawnie. Dzięki temu stosunkowo łatwo można wyłączyć część funkcji systemu bez wpływania na działanie pozostałych. Można także zrealizować kilka rozwiązań tego samego problemu i stosować je w zależności od potrzeb lub sytuacji.



Rysunek 1. Koncepcja logicznego rozdzielenia środowiska działania agentów i robotów. Przykład prezentuje wykorzystanie różnych modeli środowiska przez poszczególne agenty w systemie nawigacji robotów.

Różne agenty, odpowiedzialne za sterowanie jednym lub kilkoma robotami, mogą stosować algorytmy bazujące na różnych **modelach** robotów oraz ich środowiska. Agent realizujący algorytm nawigacji reaktywnej może postrzegać środowisko robota jako listę wartości zwracanych przez sensory odległości. Agent budujący mapę może konstruować siatkę zajętości na podstawie tych wartości. Dla agenta planującego trasę środowisko jest grafem nieskierowanym, a robot jest

skojarzony z wierzchołkiem lub krawędzią tego grafu. Schemat współpracy w takim systemie prezentuje rysunek 1. Agenty, które pracują na wyższym poziomie abstrakcji postrzegają roboty jedynie jako usługi innych agentów w systemie.

Stosowanie różnych modeli tego samego środowiska pozwala również na rozwiązanie problemu skalowalności elementów systemu. Problemy występujące w systemie wykorzystującym roboty można, ze względu na zasięg, podzielić na lokalne i globalne. Rozwiązania problemów lokalnych, dotyczących jednego robota, niewielkiej grupy robotów czy fragmentu środowiska, można skalować poprzez powielanie rozwiązań i dodawanie mocy obliczeniowej do systemu. Problemy globalne, dla których koszt obliczeniowy jest zależny od rozmiarów całego środowiska czy systemu, są znacznie trudniej skalowalne. W pracy zaproponowano rozwiązanie tego typu problemów oparte o wykorzystanie hierarchii modeli tego samego środowiska, które reprezentują różne jego fragmenty na różnych poziomach abstrakcji.

4. System nawigacji robotów mobilnych

W ramach pracy zrealizowany został prototypowy system nawigacji robotów mobilnych. Miał on na celu wykazanie, że w oparciu o zaproponowane podejście, można skonstruować system nawigacji robotów, który będzie się charakteryzował pożądanymi cechami pozafunkcjonalnymi.

Celem systemu nawigacji robotów jest zapewnienie możliwości przemieszczenia każdego z robotów tworzących system do pożądanego lokalizacji w środowisku. Przemieszczenie ma być bezpieczne dla robota, całego systemu oraz środowiska. Sposób realizacji przemieszczenia ma być optymalny, przy czym możliwe są różne kryteria optymalności przemieszczania. W prezentowanym systemie przyjętym kryterium był czas realizacji przemieszczenia.

Do zadań systemu nawigacji należy rozwiązywanie całego szeregu problemów, z których najważniejsze to: lokalizacja, modelowanie środowiska, planowanie tras i koordynacja ruchu. Problem lokalizacji nie był rozważany w zrealizowanym systemie, ponieważ sposób jego rozwiązania jest silnie związany z rodzajem wykorzystywanego sprzętu. Przyjęto, że system ma dostęp do informacji o położeniu i orientacji każdego z robotów w środowisku.

Platforma agentowa wykorzystywana w implementacji została oparta na środowisku JADE, które jest jednym z najbardziej popularnych rozwiązań tego typu. Pozwala ono na tworzenie maszyny wirtualnej łączącej wiele komputerów poprzez uruchamianie tzw. kontenerów i łączenie ich z wykorzystaniem sieci. W ten sposób tworzone jest środowisko wirtualne, w którym możliwe jest uruchamianie wielu agentów programowych. Agenty mogą być przemieszczane pomiędzy poszczególnymi kontenerami z zachowaniem stanu. JADE udostępnia agentom metody komunikacji zgodne ze standardem FIPA-ACL, oraz mechanizm Directory Faciliator, który jest rodzajem katalogu usług poszczególnych agentów.

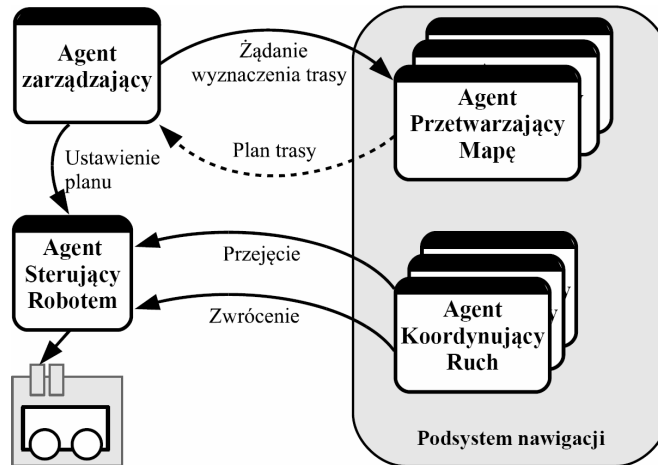
4.1. Sposób działania i architektura systemu

W zrealizowanym systemie nawigacji działa n agentów sterujących efektorami poszczególnych robotów. Implementują one algorytm nawigacji reaktywnej, który jest bezstanową metodą wyznaczania sterowania silnikami robota o napędzie różnicowym. Algorytm ten pozwala na sprawne przemieszczanie się robota w środowisku, którego kształt jest figurą wypukłą, a w którym znajdować się może pewna liczba ruchomych przeszkód (np. innych robotów).

Przemieszczanie tak sterowanych robotów w bardziej złożonych środowiskach wymaga planowania tras. W tym celu jest w systemie konstruowany i przetwarzany grafowy model środowiska, który jest rozproszony pomiędzy wiele agentów odpowiedzialnych za planowanie tras. Plan trasy to lista punktów, które kolejno musi osiągnąć robot, by dotrzeć do celu. Pomiędzy kolejnymi punktami robot może skutecznie przemieszczać się stosując algorytm nawigacji reaktywnej.

W niektórych fragmentach środowiska występować może wzmożony ruch robotów. Są to w szczególności wąskie przejścia łączące duże fragmenty środowiska. W tego typu obszarach konieczne jest stosowanie algorytmów koordynacji ruchu, które w zrealizowanym systemie są implementowane przez dedykowane do tego celu agenty. W zależności od rodzaju obszaru czy liczby robotów, agent koordynujący ruch może stosować różne algorytmy. Może przejmować pełną kontrolę nad efektorami robota lub jedynie czasowo modyfikować jego trasę.

Schemat architektury tak skonstruowanego systemu prezentuje rysunek 2.

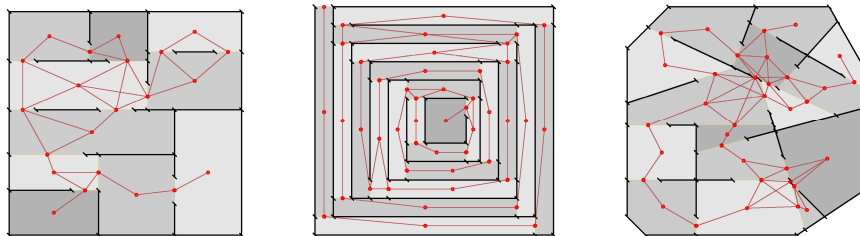


Rysunek 2. Agenty współpracujące w systemie nawigacji robotów mobilnych.

Dodatkowy agent zarządzający odpowiada za przydzielanie losowych punktów docelowych w środowisku dla robotów, które zakończyły wykonywanie poprzedniego zadania.

4.2. Mapa wielomodelowa

Mapa wielomodelowa to grafowy model środowiska działania robotów. Zrealizowany algorytm konstruowania mapy pobiera na wejściu listę współrzędnych odcinków, które reprezentują przeszkody w środowisku. Wynikiem jego działania jest zbiór pokoi, czyli fragmentów środowiska, które pokrywają je całkowicie, nie mają części wspólnych, a każdy z nich jest figurą wypukłą. Na granicach pokoi oraz w ich środkach masy umieszczane są wierzchołki grafu, a krawędzie łączą wierzchołki wewnątrz każdego z pokoi. Przykładowe wyniki działania takiego algorytmu prezentuje rysunek 3.



Rysunek 3. Przykłady efektów działania algorytmu konstruowania grafowego modelu środowiska.

Przetwarzanie scentralizowane tak utworzonego grafu jest możliwe tylko dla stosunkowo niewielkich środowisk. Istniejące algorytmy wyszukiwania tras w grafach nie są dostatecznie wydajne, by stosować je bezpośrednio do planowania tras w bardzo złożonych grafach. Dlatego w pracy zaproponowano wykorzystanie kilku poziomów abstrakcji grafowego modelu środowiska. Graf szczegółowy jest podzielony na segmenty przetwarzane przez osobne agenty, a grafy abstrakcyjne reprezentują jedynie możliwość pokonania poszczególnych segmentów.

Planowanie w takim systemie agentów wymaga komunikacji i wyznaczania kilku ścieżek w grafach na różnych poziomach abstrakcji, ale daje wyniki w znacznie krótszym czasie niż algorytmy scentralizowane.

4.3. Koordynacja ruchu

Wzrost zagęszczenia robotów w środowisku powoduje powstawanie zatorów w niektórych fragmentach środowiska. Zaproponowane w pracy podejście do projektowania systemów zarządzających robotami pozwala na łatwe wprowadzenie mechanizmów koordynujących ruch w tego typu obszarach. Utworzony w tym celu agent koordynujący ruch monitoruje sytuację w obszarze i w

miarę potrzeb przejmując sterowanie nad poszczególnymi robotami. Jego działanie ma na celu minimalizowanie średniego czasu przebywania każdego z robotów w obszarze. W ramach pracy zaimplementowane zostały dwa algorytmy koordynacji: metoda kolejkowania i metoda planowania i wykonania trajektorii.

Metoda kolejkowania nie przejmując sterowania nad efektorami robota, ale wykorzystuje algorytm nawigacji reaktywnej. Modyfikuje ona tymczasowo cele robotów, ustawiając je w kolejki robotów oczekujących na przejazd. Jednocześnie tylko jeden robot otrzymuje zezwolenie na przejazd. Metoda ta jest łatwa w implementacji i ma niskie wymagania dotyczące mocy obliczeniowej, ale może być stosowana tylko w niektórych sytuacjach, np. w koordynacji ruchu w wąskim przejściu czy w problemie kontroli dostępu do zasobu.

Metoda planowania i wykonania trajektorii wykorzystuje siatkę pokrywającą obszar w celu zaplanowania ruchu robota w środowisku tak, by nie kolidował on z przeszkodami i innymi robotami. Węzły siatki są czasowo blokowane przez przejeżdżającego robota, więc inny robot nie może w określonym czasie znaleźć się w tym samym miejscu. Trajektorie robotów są składane z elementarnych przemieszczeń pomiędzy węzłami siatki – po prostej lub po łuku. Metoda ta jest bardzo kosztowna obliczeniowo, ale pozwala na koordynowanie ruchu w obszarach dowolnego rodzaju. Porównanie skuteczności obu metod w zastosowaniu do problemu koordynacji ruchu w wąskim przejściu dało bardzo podobne wyniki.

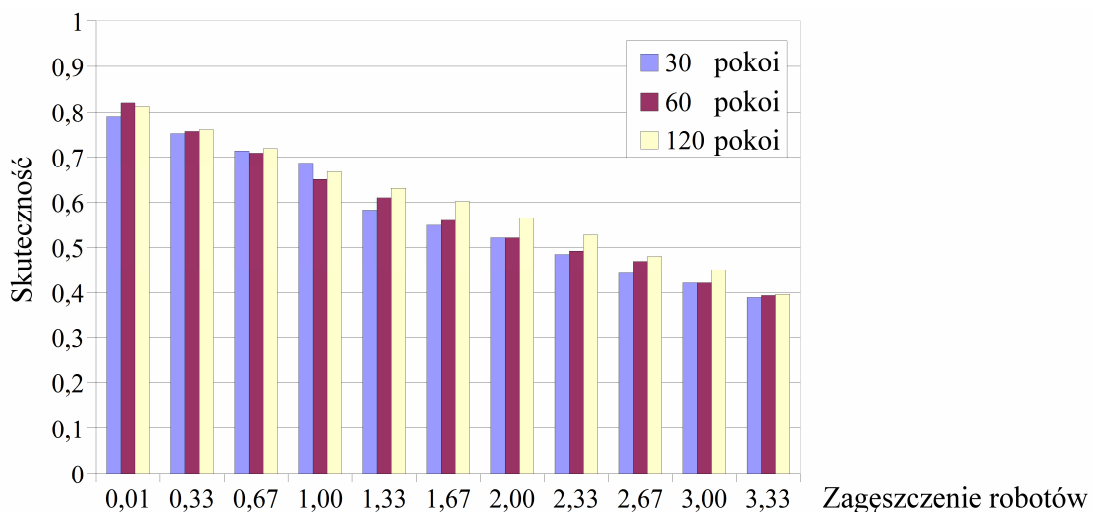
4.4. Badania własności systemu

Algorytm nawigacji reaktywnej i algorytmy koordynacji zostały przetestowane z wykorzystaniem rzeczywistych robotów. Ponieważ autor nie dysponował odpowiednio liczną grupą rzeczywistych robotów, kompletny system nawigacji został przetestowany z wykorzystaniem środowiska do symulacji robotów mobilnych RoBOSS. Miarę skuteczności działania systemu, S , zdefiniowano jako średnią skuteczność działania każdego z robotów. Skutecznością działania pojedynczego robota (s_i) był stosunek drogi pokonanej w jednostce czasu do maksymalnej drogi jaką w tym czasie mógłby pokonać jadąc z maksymalną prędkością:

$$s_i = \frac{d_i}{V_{max} * t} \quad S = \frac{\sum_{i=1}^n d_i}{V_{max} * t * n}$$

(t – czas pomiaru, V_{max} – maksymalna prędkość robota, d_i – dystans pokonany przez robota i w czasie t)

Pomiary skuteczności przeprowadzono z wykorzystaniem środowisk różnej wielkości, od 30 do 120 połączonych pokoi. W największym środowisku działały grupy od 40 do 400 robotów, czyli maksymalne zagęszczenie robotów w pokoju wynosiło ok. 3,33. Wyniki przeprowadzonych pomiarów prezentuje rysunek 4.



Rysunek 4. Zależność skuteczności działania systemu od zagęszczenia robotów w środowisku dla różnych rozmiarów środowisk

Można zauważyć, że skuteczność nie jest zależna od wielkości środowiska, ale od zagęszczenia robotów w środowisku, czyli można stwierdzić, że system jest skalowalny.

Zrealizowany system charakteryzuje się również pozostałymi własnościami pozafunkcjonalnymi, które były przedmiotem rozważań. Jego rozszerzalność została wykazana poprzez dodanie w trakcie działania systemu nowego rodzaju robota i nowego rodzaju fragmentu środowiska. Odporność na awarie przetestowano niszcząc poszczególne rodzaje agentów oraz symulując uszkodzenie robota. Każdy rodzaj awarii został wykryty przez innego agenta i obsłużony zgodnie z przyjętymi założeniami. Poszczególne elementy systemu nie są ze sobą ściśle związane. Planowanie trasy i koordynacja ruchu nie są uzależnione od rodzaju wykorzystywanych robotów, a algorytm sterujący robotem nie jest świadomy rodzaju wykonywanego przez robota zadania. Dlatego też możliwe jest wykorzystanie zaimplementowanych algorytmów w innych systemach, które będą realizowały inne zadania przy użyciu odmiennego sprzętu.

5. Wnioski

Można stwierdzić, że zaproponowane w pracy podejście do projektowania i implementowania systemów agentowych zarządzających działaniem robotów mobilnych pozwala na osiągnięcie pożądanых cech pozafunkcjonalnych. Logiczne oddzielenie agenta programowego od obiektu robota oraz jednoczesne zastosowanie wielu modeli środowiska pozwala na skonstruowanie systemu zarządzającego grupą robotów mobilnych, który jest skalowalny, rozszerzalny, zapewnia wysoką dostępność i odporność na awarie, a jego elementy mogą być z powodzeniem wykorzystywane w innych systemach.

Najciekawszym kierunkiem rozwoju zaproponowanego podejścia będą badania nad platformami agentowymi dedykowanymi do zarządzania robotami. Stworzenie takiego środowiska pozwoliłoby na szybkie i skuteczne tworzenie nowych systemów, co mogłoby zaowocować odkryciem nowych zastosowań dla grup robotów mobilnych.