Prof. dr hab. Adam Borkowski
Mgr Michał Gnatowski
Instytut Podstawowych Probemów Techniki
Polskiej Akademii Nauk

KRZEPKI MODEL WSPÓŁPRACY ZESPOŁU JEDNORODNYCH ROBOTÓW W ZADANIU PRZESZUKIWANIA POMIESZCZENIA

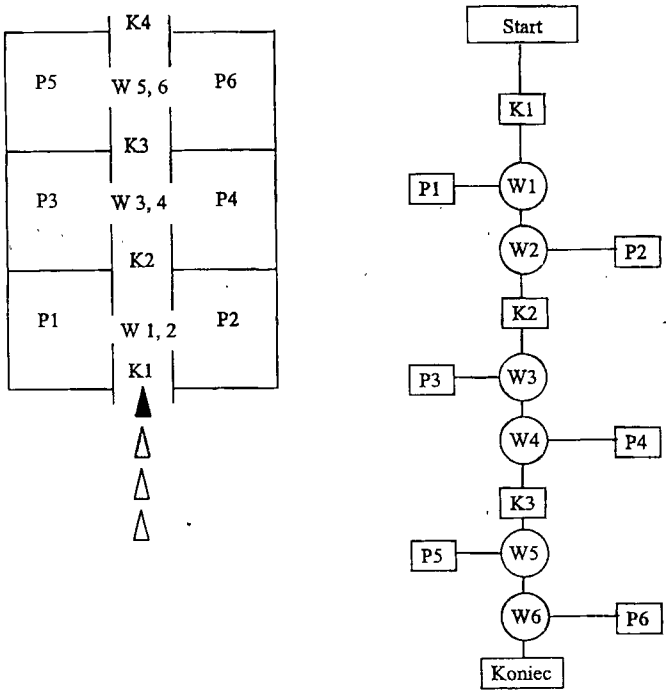
W niniejszej pracy zaproponowano model przeszukiwania pomieszczenia z wykorzystaniem zespołu robotów. Zespół robotów składa się z „leadera" oraz „zwykłych" robotów. Zakładamy, że przeszukujemy obszar z częściowo znaną mapą otoczenia (np. zburzony budynek, w celu poszukiwania ludzi). Mechanizm komunikacji ma być odporny na wszelkie sytuacje krytyczne, ma zapewniać niezawodność i być skuteczny. Zaproponowany mechanizm będzie przetestowany symulacyjnie na bazie własnego symulatora oprogramowanego w języku java.

A ROBUST COOPERATION MODEL FOR AN AREA INSPECTION

This paper presents a model for an area inspection by a team of mobile robots.The team consist of the leader and robots.The assumption is that the team has the partially known map. The communication model is to be robust and efficacious. This mechanism is to be tested with a software written in the java language.

1. WSTĘP

Zadanie przeszukiwania pomieszczenia wydaję się być ciągle aktualne. W przypadku wielkich katastrof (trzęsienia ziemi, czy ataki terrorystyczne) niezbędna jest jednocześnie duża liczba ratowników. Okazuje się, że brak ratowników jest istotnym problem w sprawnym przeprowadzaniu akcji ratowniczych. Dlatego są prowadzone badania nad wykorzystaniem robotów (ściśle zespołu robotów) do przeszukiwania gruzowisk i odnajdywania ludzi [1], [2]. Innym przykładem celowości przeszukiwania pomieszczenia przez zespół robotów, może być przypadek radioaktywnie skażonych obszarów. Zadanie przeszukania budynku jest testowane na przykładzie z rys. 1.
Celowość stosowania modelu wydaje się być oczywista. Bez modelu, każde zadanie trzeba by analizować od początku, model pozwala zagwarantować, że współpraca między robotami będzie miała miejsce, po spełnieniu określonych warunków, model pozwala ograniczyć ilość informacji przesyłanych między robotami.
Mechanizm działania jest następujący: Roboty jadą w szyku za leaderem. Po napotkaniu węzła leader wyznacza jednego z robotów do przeszukania gałęzi grafu (pokoju), po czym reszta grupy jedzie dalej. Robot po przeszukaniu pomieszczenia ma za zadanie powrócić do zespołu.



Rys. 1. Przeszukiwanie pomieszczenia: a) mapa pomieszczenia, b) graf połączeń

2. MECHANIZM KOMUNIKACJI

Bazą do niniejszego modelu komunikacji jest dobrze sprawdzony w przemyśle, protokół komunikacyjny Modbus [3]. Zaletami, które umożliwiły skuteczne stosowanie go są: prosta reguła dostępu do łącza oparta na zasadzie master – slave, zabezpieczenie przesłanych komunikatów przed błędami, potwierdzenie wykonywanych rozkazów otrzymanych zdalnie od mastera, sygnalizacja błędów, oraz wykorzystanie asynchronicznej transmisji znakowej. Istota jego polega na rozsyłaniu przez mastera rozkazów, słyszanych przez wszystkie slavy. Każdy slave ma swój adres. Jeżeli slave rozpozna, że pytanie było zaadresowane do niego, wykonuje lub odrzuca rozkaz i potwierdza, lub wysyła kod błędu. Odpowiedź slava jest słyszana przez mastera i pozostałe slavy. W klasycznym protokole Modbus slavy samoistnie nie mogą inicjować transmisji danych. Istnieje pewien zestaw funkcji używanych do komunikacji. (Np. funkcja nr 03H, oznacza pytanie o grupę rejestrów. Na takie pytanie slave odpowiada swoim adresem, kodem funkcji 03H, oraz przesyła dane, lub wysyła kod błędu, określający dlaczego nie mógł spełnić rozkazu). Na końcu ramki znajduje się suma kontrolna CRC.

W zadaniu przeszukiwania pomieszczenia odpowiednikiem mastera jest leader, odpowiednikiem slavów są pozostałe roboty z grupy.

Ramka danch pomiędzy robotami jest następująca:

```
adres, kod funkcji, słowo 1, słowo 2, crc, crc
```

gdzie słowo 1 i słowo 2 przyjmuje różne wartości w zależności od kodu funkcji (kody funkcji są omówione poniżej). W celu zapewnienia poprawności komunikacji, na końcu ramki na dwóch bajtach umieszczona jest suma kontrolna CRC.
W celu odróżnienia poszczególnych pomieszczeń przyjmuje się, że pokoje są adresowane od 1 do 1000, korytarze od 1001 do 2000, węzły od 2001 do 3000. Ponieważ na określenie pomieszczenia są przewidziane dwa bajty (czyli 2^{16}), zatem możliwości ewentualnej dalszej rozbudowy opisu grafu w ramce są ogromne.
Do komunikacji pomiędzy robotami, zdefiniowano funkcje oznaczające konkretne zadania. Funkcje używane przez leadera do wysyłania rozkazów do robotów są następujące:

1. Przeszukaj pomieszczenie, np. (5, 1, 0, 1, 0, 1, crc, crc – oznacza rozkaz przeszukania pokoju nr 1 przez robota nr 5;
2. Przeszukaj gałąź, np. (5, 2, 7, 211, 0, 4, crc, crc – rozkaz przeszukania gałęzi od węzła nr 3, do pokoju nr 4). Węzeł 3 opisujemy jako 2003;
3. Sprawdzenie obecności, np. (5, 3, 0, 8, 0, 8, crc, crc – oznacza pytanie o obecność robota nr 5. Słowa po kodzie funkcji – tutaj ósemki, przyjmują dowolne wartości);
4. Informacja o stracie robota (0, 4, 0, 5, 0, 0, crc, crc – oznacza informacje do wszystkich robotów o stracie robota nr 5);

11. Potwierdzenie obecności leadera – leader odpowiada taką ramką jakią odebrał od robota;

Funkcje używane przez roboty w komunikacji z leaderem:

1. Potwierdzenie przyjęcia rozkazu o przeszukaniu pomieszczenia i akceptacja rozkazu – jeżeli robot przyjmuje rozkaz odpowiada tak samo jaką ramkę odebrał;

129 Odmowa przyjęcia rozkazu o funkcji 1 – jeżeli robot nie może przyjąć rozkazu, w polu kodu funkcji umieszcza kod funkcji który odebrał, z jedynką na najstarszym bicie ($10000001_{bin} = 129_{dec}$);

2. Potwierdzenie przyjęcia rozkazu o przeszukaniu gałęzi i akceptacja rozkazu;

130 Odmowa przyjęcia rozkazu o funkcji 2;

3. Potwierdzenie obecności – odpowiada taką samą ramką, którą odebrał;
10. Pytanie o obecność leadera (używane przez robota o najstarszym numerze, w przypadku gdy leader zbyt długo nie komunikuje się z robotami), np. (1, 10, 0, 1, 0, 1 – oznacza pytanie robota nr 1 o obecność leadera. Słowo 1 i słowo 2 przyjmują wartości stałe – zawsze jedynki);
11. Dołączenie do grupy – po wykonaniu zadania, robot jadący torem głównym po dojechaniu do leadera i reszty grupy informuje leadera o powrocie z wykonania zadania;
20. Informacja o znalezieniu szukanego obiektu (np. człowieka przy zadaniu przeszukiwania ruin) – to w jaki sposób robot ma rozpoznać czy przeszkoda jest celem przeszukiwania danego obszaru, ani to co w takiej sytuacji robot ma robić, nie jest tematem niniejszej pracy. W celu uproszczenia przyjmuje się, że w wypadku znalezienia szukanego obiektu, robot przerywa dalsze przeszukiwanie pomieszczenia, stoi i nieustannie, co określony czas, wysyła informacje o kodzie 20 („krzyczy");

Jak widać z powyższego opisu, może się tak zdarzyć, że leader i robot jednocześnie będą wysyłały informacje. W takiej sytuacji żaden robot nie odbierze prawidłowej ramki, (gdyż suma kontrolna będzie niezgodna) i nie wyśle odpowiedzi. Robot, który wysłał pytanie i nie otrzymał odpowiedzi okresowo ponawia pytanie (po czasie będącym iloczynem stałej i

numeru robota, stąd dla każdego robota ten czas jest różny). Taki prosty mechanizm rozwiązuje konflikty jednoczesnej transmisji danych.

3. OPIS MODELU

Głównym założeniem pracy jest odporność przyjętego modelu na zakłócenia. Wymaga to następujących czynności:

- Rozbudowania protokołu komunikacji o mechanizmy samokontroli;
- Zaproponowania takiego algorytmu przeszukiwania, żeby strata jednego robota nie uniemożliwiała dalszej pracy grupy;
- Możliwość zastąpienia funkcji leadera przez innego robota, w przypadku zniszczenia leadera;

Algorytm przeszukiwania pomieszczenia jest następujący:

- Leader znając mapę w postaci grafu (rys. 1.b) znajduje drogę od startu do celu z największa ilością węzłów (tor główny);
- Po dojechaniu do węzła wyznacza robota (z końca szyku) do przeszukania gałęzi. Jeżeli wewnątrz gałęzi są inne węzły, wówczas leader wysyła tyle robotów, ile jest liści gałęzi. Jeżeli jedzie więcej niż jeden robot, leader wyznacza leadera podzespołu;
- Robot lub roboty po przeszukaniu gałęzi wracają do węzła, w którym oddzieliły się od leadera i jadą torem głównym, aż do napotkania leadera;
- Leader po wysłaniu całej grupy do przeszukiwania gałęzi stoi i czeka na pierwszego robota, który wróci z ukończonego zadania. Jeżeli taki robot się pojawi, leader kontynuuje jazdę torem głównym;

Miarą skuteczności algorytmu może być stosunek czasu przeszukania pomieszczenia przez zespół robotów, do wyliczonego czasu przeszukiwania obszaru przez jednego robota. Czas przszukiwania obszaru przez jednego robota jest równy stosunkowi długości trasy do średniej prędkości robota (w modelu symulacyjnym zakładamy, że prędkość robota jest stała). Skuteczność algorytmu przedstawia poniższy wzór:

$$\xi = T_{zes} / T_{rob} = T_{zes} * v / s \qquad (1.1)$$

gdzie:
T_{zes} – czas wykonania zadania przez cały zespół (od startu do momentu aż wszystkie roboty nie dotrą do celu);
T_{rob} – wyliczony czas wykonania zadania przez pojedyńczego robota;
v – prędkość robota (zakładamy, że jest stała);
s – całkowita droga przeszukiwanego obszaru;

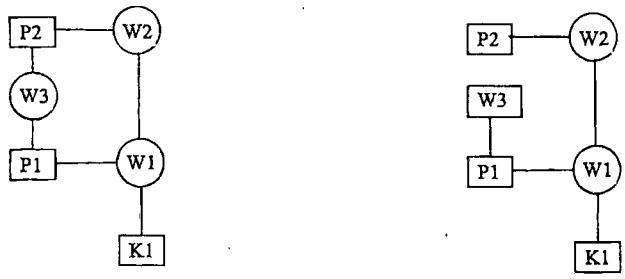
Z powyższego wzoru widać, że jeżeli skuteczność algorytmu $\xi < 1$, to oznacza, że cały zespół wykonuje zadanie szybciej niż zrobiłby to pojedyńczy robot. Natomiast $\xi > 1$, oznacza dłuższy czas wykonywania zadania niż pojedyńczy robot.
W trakcie wykonywania zadania roboty mogą znajdować się w określonych trybach pracy. Celem modelu jest rozpoznanie wszystkich możliwych stanów i możliwych przejść pomiędzy tymi stanami. I tak leader przyjmuje następujące tryby pracy:

I. Szukanie toru głównego (czyli drogi od startu do celu, przez maksymalną ilość węzłów), z redukcją zbędnych węzłów. Przykład redukcji zbędnych węzłów przedstawia rys. 1.2;
II. Dojazd do węzła (w trakcie jazdy leader przeszukuje drogę, którą jedzie);

III. Opracowanie trasy w gałęzi bocznej i wysłanie tam robota z końca szyku;
IV. Oczekiwanie na roboty (gdy wszystkie zostaną wysłane do gałęzi bocznych);
V. Zakończenie zadania;

„Zwykły" robot przyjmuje następujące tryby pracy:

I. Podążanie za leaderem;
II. Przeszukiwanie pomieszczenia (algortym przeszukiwania pomieszczenia nie jest tematem niniejszej pracy)
III. Powrót do węzła początkowego gałęzi;
IV. Dojazd do leadera wzdłóż toru głównego;
V. Dołączenie do grupy;



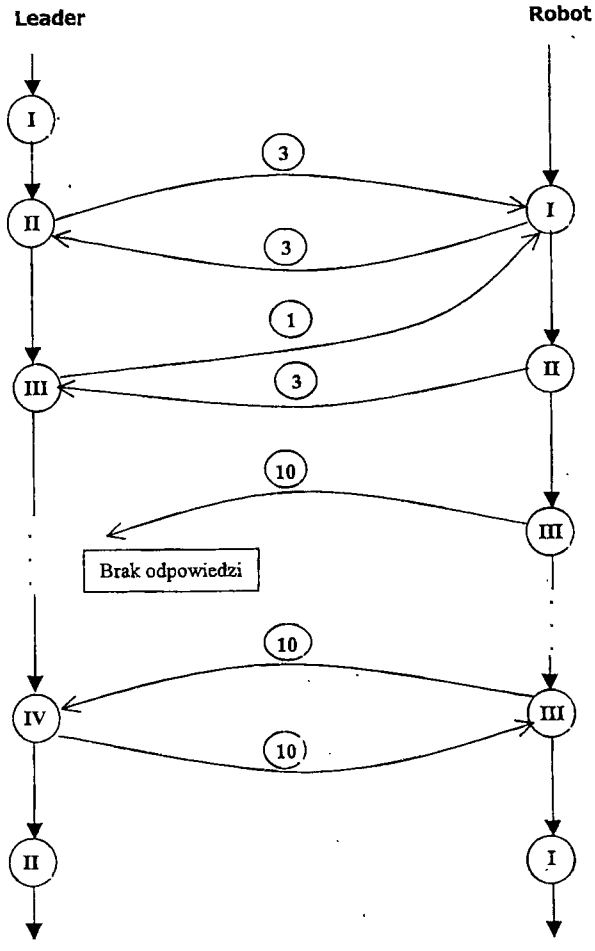
Rys. 2. Mechanizm redukcji węzłów. a) graf przed redukcją; b) graf po redukcji

W trakcie realizacji zadania występuje następująca wymiana informacji pomiędzy robotami:

- Leader w trybie pracy I, robot w trybie pracy I, po dojechaniu do węzła, np. 1, leader wysyła robota z końca szyku do przeszukania pomieszczenia nr 1;
- Robot jedzie do pomieszczenia nr 1, przeszukuje, po czym wraca do węzła głównego gałęzi i jedzie dalej torem głównym aż do momentu znalezienia reszty zespołu. W tym czasie okresowo wysyła informacje o obecność leadera (z kodem funkcji 10);
- W tym czasie leader z resztą grupy jedzie dalej. W trakcie jazdy leader okresowo odpytuje poszczególne roboty o ich obecność w grupie (kod funkcji 3). Jeżeli któryś robot kilkakrotnie (ustaloną ilość razy) nie odpowie na pytanie, leader uzna go za straconego i poinformuje o tym resztę grupy (kod funkcji 4). Ta informacja jest potrzebna robotom, ponieważ przyjmuje się, że w wypadku straty leadera, funkcję leadera przujmuje robot o najniższym numerze;
- Jeżeli w czasie jazdy robot o najniższym numerze przez określony czas nie otrzyma pytania od leadera o obecność, sam zada pytanie o obecność leadera (kod funkcji: 10). Podobnie jak w powyższym punkcie, jeżeli leader nie odpowie ustaloną ilość razy, robot o najniższym numerze uzna leadera za utraconego, sam siebie mianuje leaderem i poinformuje o tym resztę grupy); Taki sam mechanizm zastosuje robot o wyższym numerze od poprzedniego, lecz czas oczekiwania na pytanie będzie dłuższy. Ma to zastosowanie w wypadku, gdyby jednocześnie zginął leader i robot nr 1;

Przykładowy schemat przełączania pomiędzy poszczególnym trybami pracy jest pokazany na rys. 3. Na początku realizacji zadania leader szuka toru głównego od startu do celu (działa w

trybie I). Następnie przełącza się w tryb II i rozpoczyna realizację wzdłóż toru głównego. Robot jest w trybie pracy I i podąża za leaderem. W tym czasie leader okresowo sprawdza obecność robota (kod funkcji 3) i robot potwierdza swoją obecność (kod funkcji 3). Po dojechaniu do węzła leader przełącza się w tryb III, opracowuje trasę dla robota (sprawdza ile robotów trzeba wysłać do gałęzi) i wysyła robota do przeszukania gałęzi (kod funkcji 1). Leader przełącza się w tryb II, i kontynuuje jazdę do kolejnego węzła z pozostałymi robotami z grupy (na rys. 3, pominięto komunikację z innymi robotami). Tak się dzieje aż leader nie wyśle ostatnigo robota do przeszukania gałęzi. Wtedy leader przełącza się w tryb IV i czeka na któregoś robota. Robot po otrzymaniu polecenia przeszukania pomieszczenia przełącza się w tryb III, oddziela się od zespołu i przeszukuje zadaną gałąź. Po zakończeniu przeszukiwania powraca do gałęzi w której oddzielił się od zespołu i kontunuuje trasę torem głównym. W tym czasie próbuje odnaleść leadera (kod funkcji 10). Otrzymanie potwierdzenia obedności leadera powoduje, że robot przełącza się w tryb I (podążanie za lederem).



Rys. 3. Przykładowa wymiana informacji pomiędzy leaderem a robotem, oraz przejścia pomiędzy trybami pracy leadera i robota.

4. OPIS CZĘŚCI SYMULACYJNEJ

Sposób działania symulatora jest oparty na działaniu serwera do rozgrywania meczów piłki nożnej (RoboCup) [4]. Idea rozgrywania symulacyjnych meczów piłkarskich polega na tym, że serwer będący jednocześnie boiskiem i sędzią, co ustalony czas (100ms) wysyła do każdego zawodnika (czyli do każdego klienta) informacje o otaczającym go świecie. Informacja przesyłana do klienta zawiera: znaczniki widziane przez gracza, informacje o położeniu i prędkości piłki (jeżeli jest w zasięgu „wzroku" gracza), informacje o innych widzianych graczach itp. Również co ustalony czas, zawodnik może wysyłać do serwera informacje o podejmowanej czynności np. bieg, obrót, kopnięcie piłki, wysłanie komunikatu itp. Komunikacja pomiędzy graczami odbywa się za pomocą komend. Jeżeli zawodnik wysyła informację do innych zawodników, informuje serwer o treści swojej komendy. W następnym kroku serwer wszystkim zawodnikom znajdującym się w określonym promieniu od krzyczącego zawodnika przesyła informację o treści komunikatu.
Bazując na powyższym opisie, zakładamy, że roboty to są klienci, którzy logują się do środowiska – serwera. Co ustalony czas występuje wymiana informacji pomiędzy serwerem a klientami. Serwer wysyła dane o otaczającym świecie (przeszkody, przesył danych itp.) a roboty – klienci wysyłając dane do serwera uaktualniają swoją pozycję, wysyłają dane do innych robotów itp.

5. WNIOSKI

W artykule przedstawiono mechanizm przeszukiwania pomieszczenia przez zespół jednorodnych robotów. Zaletą tego algorytmu jest odporność na zakłócenia spowodowane zniszczeniem robota, ograniczonym promieniem zasięgu komunikacji, itp. Autor zdaję sobie sprawę, że zagadnienie przeszukiwania pomieszczenia jest w ogólnym przypadku skomplikowane i zaproponowany mechanizm nie rozwiąże wszystkich problemów. Niniejsza praca może być podstawą do dalszych badań. Propozycje rozbudowy systemu są następujące:

- Nie omówiono, co ma się dziać gdy robot znajdzie szukany cel, ani jak ma go rozpoznać. To zagadneinie jest oddzielnym tematem i zależy od konkretnej implementacji modelu;
- Nie omówiono jaki ma być algorym przeszukiwania pomieszczenia. To zagadnienie należy rozpatrywać oddzielnie od niniejszego modelu;
- Obecnie lokalizacja robotów jest realizowana poprzez idealną odometrię. Warto byłoby opracować inny mechanizm ułatwiający skuteczną i praktycznie realizowalną lokalizację robota;
- Zakłada się, że roboty są jednorodne. Ułatwia to możliwość redundancji robotów w wypadku zniszczenia jednego z nich. Ciekawym rozwinięciem niniejszej pracy byłoby opracowanie podobnego mechanizmu dla różnorodych robotów;
- Nie omówiono mechanizmu wyznaczania toru głównego, ani mechanizmu redukcji węzłów;
- Obecnie zakłada się, że robot albo jest sprawny, albo jest zniszczony. W proponowanym mechaniźmie nie ma możliwości zgłaszania usterek, które ograniczają tylko częściowo możliwości robota.
- Nie przedstawiono mechanizmu korekcji mapy;

LITERATURA

[1] Hiroaki Kitano, Satoshi Tadokoro: *RoboCup Rescue. A Great Challenge for Multiagent and Intelligent System*; American Association for Artificial Intelligence. 0738-4602-2001

[2] Alan C. Schultz: *The 2000 AAAI Mobile Robot Competition and Exhibition, American Association for Artificial Intelligence;* 0738-4602-2001

[3] Modicon: *Modicon Modbus Protocol Reference Guide; PI-MBUS-300 Rev.D. Modicon, Inc. 1992*;

[4] E. Corten, K.Dorer, R.Heintz: *Soccerserver Manual Ver. 5 Rev. 00 beta, 1999*;