

dr inż. Waldemar Małopolski
Politechnika Krakowska

METODA WYZNACZANIA DOWOLNYCH TRAS PRZEJAZDU OBIEKTÓW SYSTEMU TRANSPORTOWEGO W ŚRODOWISKU SYMULACYJNYM ARENA

W artykule przedstawiono model systemu transportowego, zbudowany w środowisku symulacyjnym Arena, który umożliwia dowolne wyznaczenie trasy przejazdu obiektu systemu transportowego. Opisano również sposób integracji modelu z aplikacją zewnętrzną, co umożliwia wykorzystanie Areny do testowania algorytmów sterujących systemem transportowym.

A METHOD OF DETERMINATION OPTIONAL ROUTE OF TRANSPORTATION SYSTEM OBJECTS IN ARENA SIMULATION ENVIRONMENT

This paper presents a transportation system model, developed in Arena simulation software, which enables optional determination of route of transportation system objects. In this paper an integration of model with external desktop applications is presented too. Therefore Arena can be used for testing transportation system control algorithms.

1. WPROWADZENIE

Rozwój gospodarki opartej na zasadach rynkowych wymusza ciągle wprowadzanie ulepszeń we wszystkich obszarach działania związanych z wytwarzaniem dóbr konsumpcyjnych. Głównym celem tych działań jest sprośowanie oczekiwaniom klientów. Wymusza to konieczność dostosowania oferowanych produktów do gustów klientów oraz obniżenia ich ceny i podniesienia jakości. Pozycja producentów na rynku, a nawet ich przetrwanie, zależy od umiejętności szybkiego dostosowania się do zmieniających się potrzeb. Zdolność do szybkiego i taniego wprowadzania zmian w zakresie technologii i organizacji produkcji staje się zagadnieniem kluczowym. Z tego względu dużego znaczenia nabierają wszelkie metody i narzędzia wspomagające poszukiwanie najlepszych (optymalnych) rozwiązań.

Zagadnienia optymalizacyjne mogą być rozwijane metodami analitycznym i lub symulacyjnymi. W przypadku braku rozwiązań analitycznych, bardzo często jedynym możliwym rozwiązaniem jest zastosowanie metod symulacyjnych. Stało się to przyczyną dynamicznego rozwoju narzędzi symulacyjnych ukierunkowanych na różne obszary zastosowań. Podstawową zaletą stosowania symulacji jest możliwość poszukiwania najlepszych rozwiązań bazując na matematycznym modelu obiektu (systemu) rzeczywistego. Pozwala to na obniżenie kosztów takich badań a często zapobiega uszkodzeniom obiektu rzeczywistego.

W sferze produkcyjnej symulacja komputerowa znalazła szerokie zastosowanie do rozwiązywania problemów zarządzania i sterowania produkcją. Narzędzia symulacyjne są wykorzystywane do optymalizacji budowy linii produkcyjnych, podsystemów, a nawet całych systemów produkcyjnych, czy też fabryk. Możliwe jest też wykorzystanie symulacji

komputerowej do testowania nowych rozwiązań w zakresie metod sterowania systemami produkcyjnymi. Pozwala to na weryfikację poprawności przyjętych rozwiązań przed wdrożeniem ich do obiektu rzeczywistego.

Na przestrzeni ostatnich lat obserwuje się dynamiczny rozwój badań naukowych nad zastosowaniem koncepcji sterownia rozproszonego (agentowego) do sterowania systemami produkcyjnymi. W Instytucie Technologii Maszyn i Automatyzacji Produkcji (ITMiAP) Politechniki Krakowskiej jest rozwijany oryginalny wieloagentowy system sterowania produkcją AIM (ang. *Agents Integrated Manufacturing*) [7, 11]. Ze względu na to, że w procesie produkcyjnym bardzo duże znaczenie odgrywa transport międzyoperacyjny, podjęto obecnie w ITMiAP prace związane z zastosowaniem wieloagentowego sterownia rozproszonego do podsystemu transportowego [9]. Ponieważ czynności transportowe nie generują wartości dodanej, należy dążyć do minimalizacji ich kosztów. Stawia to przed systemem sterującym szereg trudnych zadań. Problemy te są obiektem badań w wielu ośrodkach naukowych na świecie.

Podstawowa grupa zagadnień jest związana z techniczną budową podsystemu transportowego, a szczególnie z budową obiektów wchodzących w jego skład. Ze względu na dużą elastyczność w zakresie realizacji zadań transportowych, często stosowanym rozwiązaniem są autonomiczne wózki mobilne. Z tego też względu podjęto w ITMiAP prace na budowę dwóch autonomicznych wózków mobilnych [9]. Autonomiczność wózków wymaga zastosowania odpowiednich rozwiązań w zakresie zasilania, sterowania napędem, komunikacji z systemem sterującym i lokalizacji położenia wózka [6]. Zbudowane wózki umożliwią weryfikację opracowanego systemu sterowania.

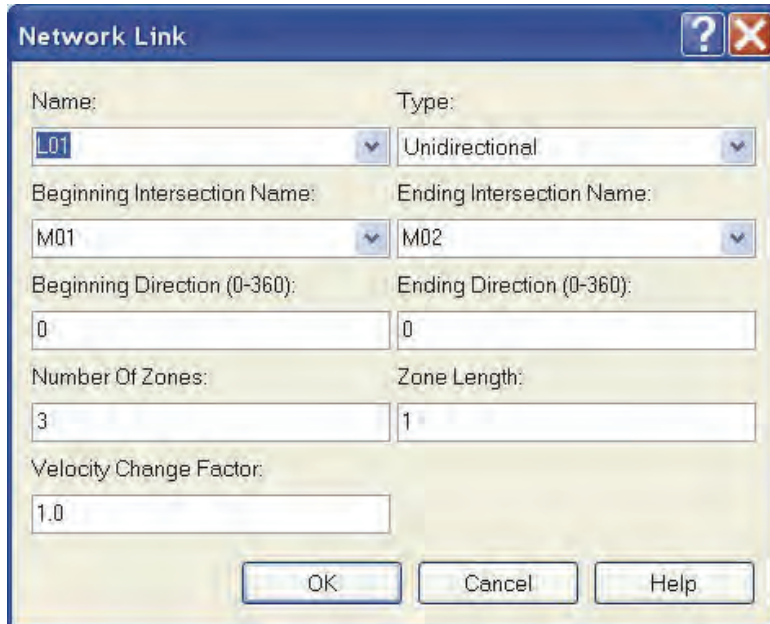
Druga grupa zagadnień, wymagających rozwiązania, jest związana z budową samego systemu sterowania. Do ważniejszych problemów, które muszą być rozwiązane, można zaliczyć wyznaczanie tras przejazdu wózków [2, 3], harmonogramowanie zadań transportowych [5], czy też rozwiązania zapewniające efektywne wykorzystanie wózków, w tym zapobieganie kolizjom [1, 4] i blokadom systemu transportowego [8].

Zbudowanie systemu sterującego podsystemem transportowym jest zatem zagadnieniem złożonym i wymaga dokładnego przetestowania przed wdrożeniem go do obiektu rzeczywistego. Z tego względu celowym wydaje się opracowanie modelu symulacyjnego podsystemu transportowego, który pozwoli na testowanie i weryfikowanie różnych rozwiązań zaimplementowanych do systemu sterującego. Pociągająco to jednak szereg wymogów, jakie musi spełnić symulator. Podstawowym wymogiem jest możliwość wymiany danych (integracji) systemu sterującego z modelem w czasie rzeczywistym. Drugi ważny wymóg, to możliwość dokładnego modelowania sieci transportowej i możliwość dowolnego wyznaczania tras przejazdu wózków.

2. MODELOWANIE SYSTEMÓW TRANSPORTOWYCH W ŚRODOWISKU SYMULACYJNYM ARENA

Arena jest jednym z najbardziej popularnych programów komercyjnych do modelowania i symulacji systemów dyskretnych. Interfejs graficzny umożliwia bardzo łatwe i szybkie budowanie modeli i przeprowadzanie symulacji. Bogaty zestaw narzędzi wspomagających przygotowanie danych wejściowych oraz analizę wyników symulacji w znaczący sposób ułatwia pracę w tym środowisku. Na szczególną uwagę zasługuje możliwość wymiany

danych pom iędzy m odelem sym ulacyjnym a aplikacj ą zewn ętrzną w trakcie procesu symulacji w czasie rzeczywistym . Dzi ęki t emu mo żna wykorzysta ć Aren ę do sterowania obiektami rzeczywistymi lub zbudowa ć w Arenie model systemu rzeczywistego i testowa ć na nim układy lub programy steruj ące.



Rys. 1. Definiowanie linii (dróg) transportowych



Rys. 2. Definiowanie sieci transportowej

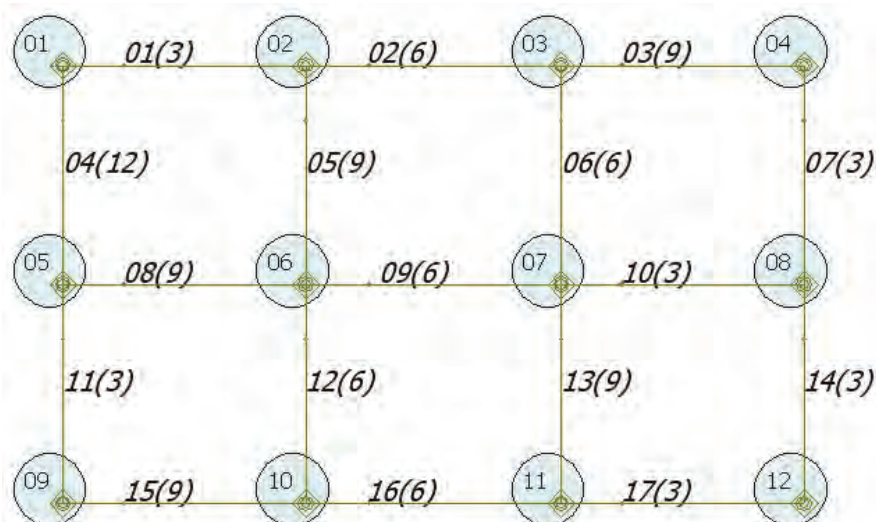


Rys. 3. Definiowanie obiektów systemu transportowego

W zakresie modelowania systemów transportowych Arena dostarcza bogaty zestaw narzędzi, który pozwala na modelowanie systemów zbudowanych z obiektów dyskretnych, np. robotów mobilnych, jak również z systemów transportu ciągłego, np. taśmociągów. Modelowanie dyskretnych systemów transportowych można zrealizować na dwa sposoby.

Pierwszy sposób polega na uproszczonym definiowaniu dróg transportowych, po których porusza się obiekt. Sprowadza się to do definiowania długości dróg transportowych i prędkości, z jaką realizowany jest transport. Cała logika związana z zapobieganiem powstawaniu zastoju oraz nie dopuszczanie do kolizji musi być zawarta w modelu. Takie podejście jest w wielu prostych przypadkach zupełnie wystarczające.

Drugi sposób modelowania dyskretnych systemów transportowych daje większe możliwości w zakresie definiowania parametrów systemu transportowego. Podstawową cechą tego podejścia jest możliwość definiowania sieci transportowej, składającej się z węzłów (skrzyżowań) i linii (dróg transportowych), która dokładnie określa przestrzeń transportową, w której mogą się poruszać obiekty systemu transportowego. Definiowanie sieci transportowej wymaga określenia wszystkich linii łączących węzły, rys. 1. Oprócz podania nazwy linii, nazwy węzła początkowego i końcowego oraz ich kierunków (od 0 do 360 stopni), definiowany jest typ linii (droga jednokierunkowa, dwukierunkowa i kołowa) oraz liczba odcinków z których składa się dana linia. Określenie kierunku linii przy węźle początkowym i końcowym pozwala na definiowanie zakrętów, co wpływa na zmniejszenie prędkości, z jaką porusza się obiekt, który pokonuje zakręt. Zdefiniowanie długości odcinka jednoznacznie określa długość linii. Ponadto definiowany jest współczynnik zmiany prędkości jazdy obiektu transportowego na danej linii w stosunku do jego prędkości nominalnej.

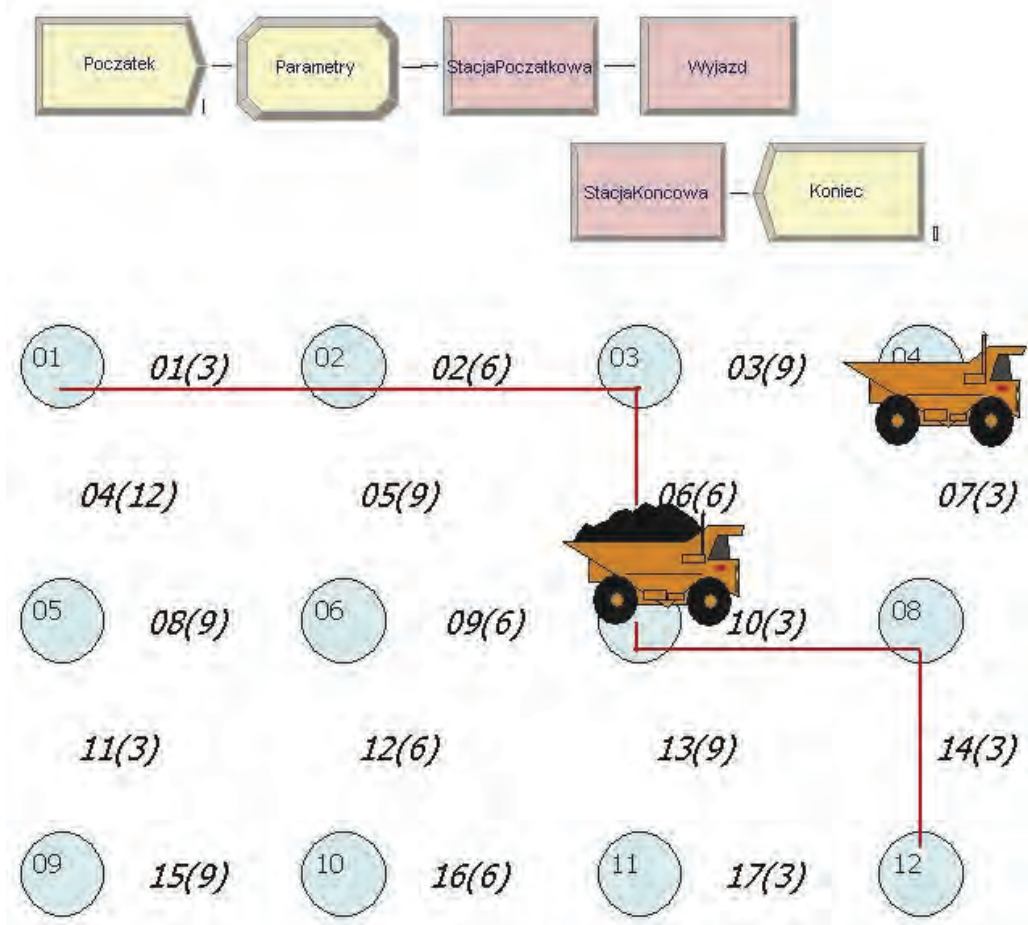


Rys. 4. Graficzna prezentacja sieci transportowej

Zbiór wszystkich linii (dróg transportowych) jest wykorzystywany do definiowania sieci transportowej, rys. 2. W modelu można zdefiniować wiele różnych sieci transportowych. Po każdej sieci mogą poruszać się inne obiekty transportowe. Definiowanie obiektów transportowych polega m.in. na wprowadzeniu następujących danych: nazwy obiektu (nazwy grupy obiektów), liczby obiektów w danej grupie, nazwy sieci transportowej, po której mogą się poruszać obiekty, maksymalnej prędkości, przyspieszenia, opóźnienia, współczynnika zmiany prędkości na zakrętach oraz miejsca początkowego położenia obiektu w sieci, rys. 3.

Po zdefiniowaniu sieci transportowej i obiektów poruszających się po danej sieci można przystąpić do utworzenia graficznej wizualizacji sieci, która jest niezbędna do animacji działania systemu transportowego. W tym celu rysuje się odcinki reprezentujące linie, które łączą się w węzłach. Po tych odcinkach będą się poruszały obiekty systemu transportowego podczas realizacji swoich zadań. Na rys. 4 przedstawiono sieć transportową. Dodatkowo zaznaczono węzły sieci i podano ich numery. Na każdej linii umieszczono informację o jej numerze i podano w nawiasie długość drogi. Kolejnym etapem jest rozbudowanie modelu symulacyjnego o elementy wykorzystujące system transportowy.

Na rys. 5 przedstawiono schemat prostego modelu, realizującego zadanie transportowe, polegające na przewiezieniu przedmiotu z węzła nr 1 do węzła nr 12. Model składa się z kilku modułów. Pierwszy moduł „Początek” generuje przewożony przedmiot, który przechodzi do modułu „Parametry”. W module tym zostaje do parametru „NumerWozka” przypisana wartość 1, która określa pierwszy wózek z dwulem entowego zbioru obiektów systemu transportowego. Kolejnym modułem jest „StacjaPoczątkowa” typu Station, w którym przedmiotowi zostaje przypisane aktualne położenie. Zadania transportowe w modelu mogą być realizowane pomiędzy stacjami zdefiniowanymi w modułach typu Station. Każda stacja jest przypisana do określonego węzła sieci transportowej. Dzięki temu definiując zadanie transportowe od stacji do stacji określamy węzeł początkowy i końcowy. W omawianym przykładzie „StacjaPoczątkowa” jest przypisana do węzła nr 1.



Rys. 5. Trasa przejazdu obiektu systemu transportowego wg kryterium najkrótszej drogi

Kolejnym modelem, do którego wchodzi przedmiot jest model „Wyjazd”, który odpowiada za przybycie wózka nr 1 do węzła pierwszego, zajęcie wózka przez przedmiot i rozpoczęcie transportu przedmiotu na wózku do węzła nr 12, czyli do stacji „StacjaKoncowa”. Po dotarciu do węzła nr 12, w modelu „StacjaKoncowa” następuje zwolnienie wózka przez przedmiot i opuszczenie modelu.

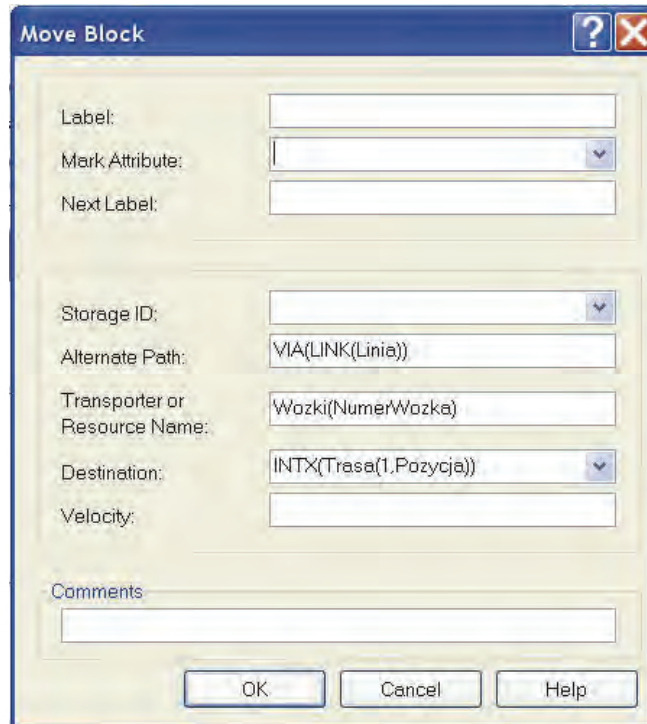
Zadanie transportowe zostało wykonane. Pojawia się jednak pytanie, w jaki sposób Arena dokonała wyboru trasy w sieci transportowej (trasa zaznaczona czerwoną linią). Po zdefiniowaniu sieci transportowej z długościami dróg Arena buduje wewnętrzną macierz możliwych połączeń pomiędzy węzłami ze względu na najkrótszą odległość. W trakcie symulacji obiekt systemu transportowego jest przemierzany najkrótszą drogą od węzła początkowego do końcowego. Jedynym kryterium wyboru jest minimalna długość trasy. Bardzo często może zdarzyć się tak, że czas przejazdu inną trasą (dłuższą) będzie krótszy. Zależy to od maksymalnych prędkości, jakie obiekt systemu transportowego może rozwinąć na poszczególnych liniach sieci transportowej. Pewnym ułatwieniem wpływającym na wybór trasy jest możliwość zdefiniowania pośredniego miejsca w sieci transportowej, przez które obiekt ma przejechać, jadąc od węzła początkowego do końcowego. Wymaga to jednak stosowania specjalnych modeli, a i tak nie daje nam pełnej kontroli nad wyborem trasy przejazdu. Chcąc zastosować inne kryteria wyboru trasy przejazdu dla danego obiektu systemu transportowego należy zastosować rozwiązania niestandardowe. Jest to szczególnie istotne, gdy celem budowy modelu systemu transportowego jest testowanie zewnętrznego systemu sterowania.

3. METODA WYZNACZANIA DOWOLNYCH TRAS PRZEJAZDU OBIEKTÓW SYSTEMU TRANSPORTOWEGO

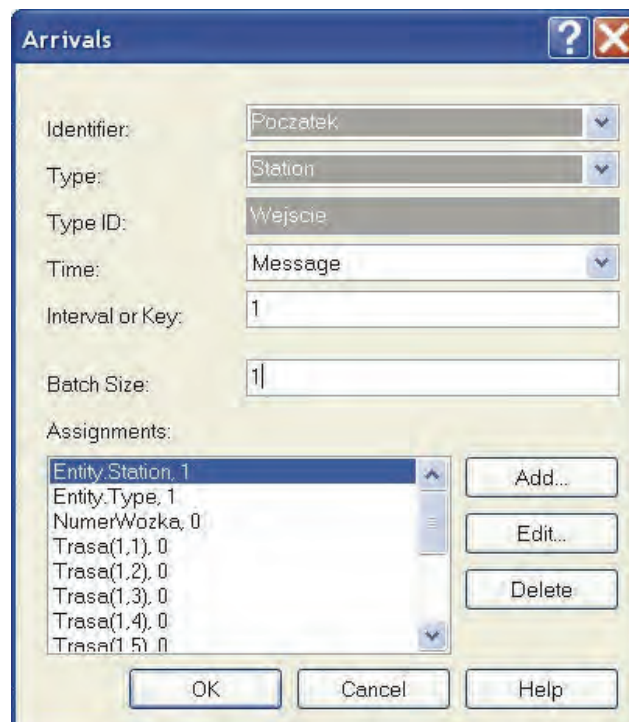
Przedstawiona poniżej metoda jest rozwiązaniem niestandardowym. Podstawowym założeniem przyjętym przy opracowywaniu tej metody była możliwość dowolnego wyznaczenia tras przejazdu obiektów systemu transportowego. W tym celu zbudowano model w programie Arena wykorzystujący specjalny blok „Move”, rys. 6, który pozwala na wskazanie miejsca pośredniego, przez które ma przebiegać trasa przejazdu. Ponadto trasę przejazdu od węzła początkowego do końcowego definiuje się przez podanie wszystkich pośrednich węzłów. Realizacja złożonego zadania transportowego została sprowadzona do realizacji elementarnych zadań transportowych pomiędzy sąsiadującymi węzłami przez linię łączącą oba węzły. Linia łącząca sąsiadujące węzły jest określona przez wyrażenie „VIA(LINK(Linia))”. Węzeł docelowy (w kolejnym elementarnym zadaniu transportowym) jest określony przez wyrażenie „INTX(Trasa(1,Pozycja))”. Takie rozwiązanie zapewnia pełną kontrolę nad trasą przejazdu obiektu od węzła początkowego do końcowego.

Drugim ważnym założeniem, mającym wpływ na przyjęte rozwiązanie, było umożliwienie wymiany danych z aplikacją zewnętrzną, co pozwoliłoby na zbudowanie w Arenie wirtualnego środowiska do testowania zewnętrznego systemu sterowania obiektami i systemu transportowego. W tym celu wprowadzono do modelu specjalny element „Arrivals”, który umożliwia przesyłanie danych do Areny. Element ten w oparciu o otrzymane dane generuje nowe jednostki (reprezentujące zadania transportowe lub przedmioty, które mają być transportowane) i wprowadza je do modelu typu Station. Do tak wygenerowanych jednostek element „Arrivals” dodaje szereg informacji (otrzymanych z aplikacji zewnętrznej), opisujących

zadaną trasę przejazdu, rys. 7. Dane te są zapisywane w postaci wartości atrybutów przynależących do konkretnych jednostek w następującej kolejności: numer stacji (miejsce, gdzie pojawiają się jednostki), numer przedmiotu transportowanego, numer obiektu (transportującego przedmiot) i kolejne numery węzłów zdefiniowanej trasy, jako wartości tablicy.

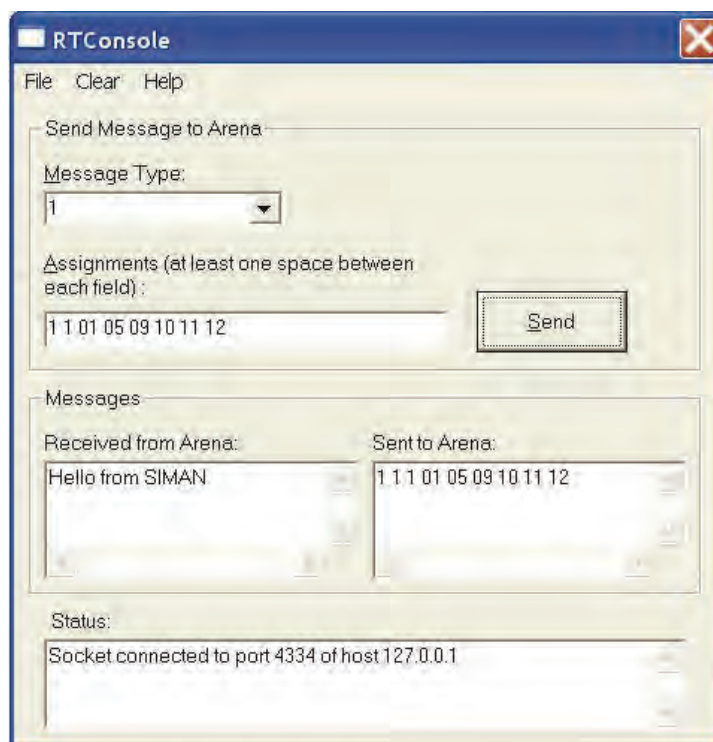


Rys. 6. Definiowanie elementarnej drogi przejazdu obiektu systemu transportowego



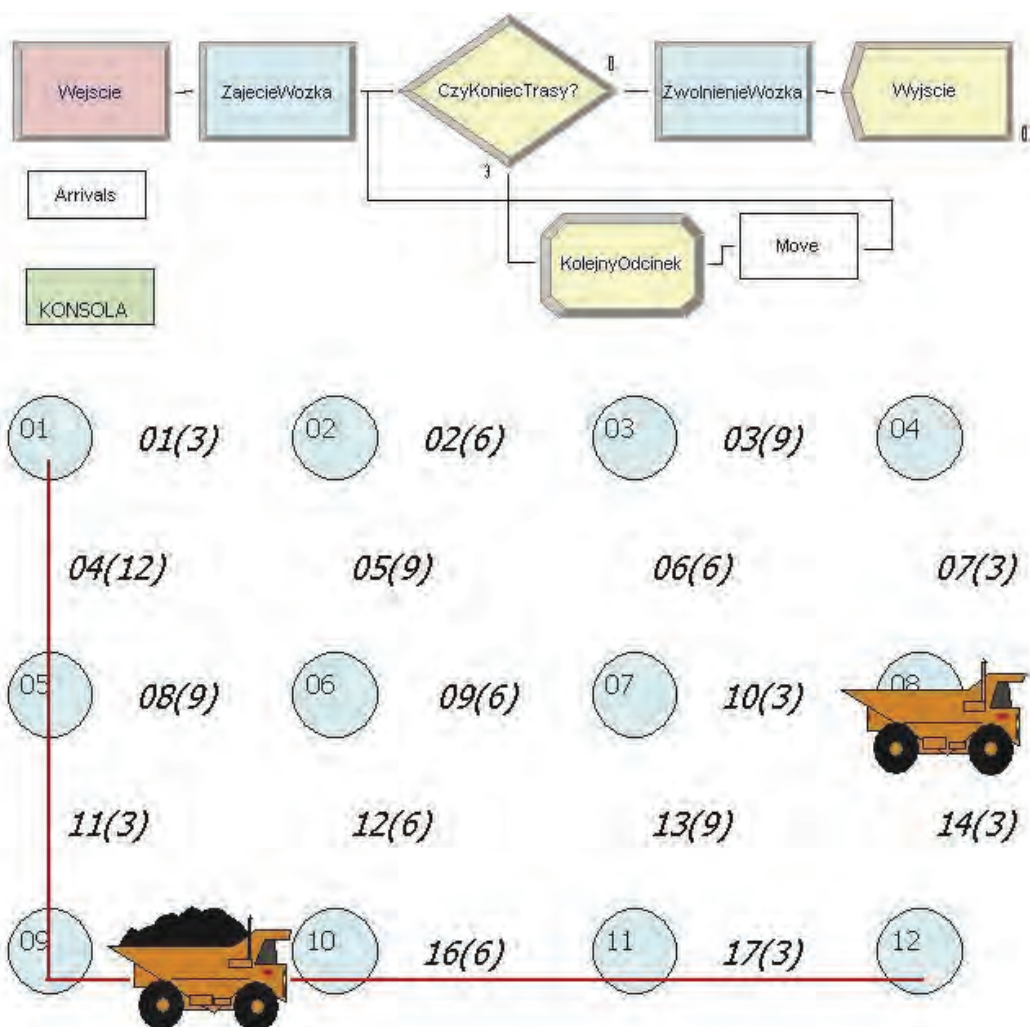
Rys. 7. Integracja modelu z aplikacją zewnętrzną

Dane określające trasę przejazdu są przesyłane z aplikacji zewnętrznej o nazwie „RT-Console” (rys. 8). Aby umożliwić przesyłanie danych, konieczne jest połączenie Areny w tryb pracy w czasie rzeczywistym. W tym trybie Arena „nasłuchuje” na porcie 4334 i jest gotowa do nawiązania połączenia. Ponadto czas „symulacyjny” w programie Arena jest zgodny z czasem komputera. Po nawiązaniu połączenia można przygotować w aplikacji RT-Console dane o zadanej trasie przejazdu. Kolejność danych została opisana powyżej. Ciąg wartości „01 05 09 10 11 12” określa kolejne numery węzłów, składających się na zadaną trasę przejazdu.



Rys. 8. Aplikacja zewnętrzna wyznaczająca trasę przejazdu

Po wysłaniu danych do Areny, zostają one odebrane przez element „Arrivals” i wygenerowana jednostka (opisana powyżej) przechodzi do modułu „ZajacieWozka” (rys. 9). W module tym zostaje jednostce przydzielony wózek i jednostka przechodzi do modułu „CzyKoniecTrasy?”, gdzie następuje sprawdzenie osiągnięcia miejsca docelowego. Jeżeli wózek dojechał do miejsca docelowego, to następuje zwolnienie wózka przez jednostkę w module „ZwolnienieWozka” i opuszczenie modułu. W przeciwnym wypadku jednostka przechodzi do modułu „KolejnyOdcinek”, w którym są przygotowywane dane potrzebne do rozpoczęcia elementarnego przejazdu wywołanego przez blok „Move”. Po pokonaniu pierwszego odcinka cykl się powtarza, aż do osiągnięcia węzła końcowego trasy. W ten sposób trasa przejazdu obiektu (zaznaczona czerwoną linią na rys. 9) jest zgodna z trasą zadaną w aplikacji RTConsole.



Rys. 9. Trasa przejazdu obiektu systemu transportowego zdefiniowana w sposób dowolny

W trakcie realizacji zadania transportowego uwzględniane są parametry mające wpływ na prędkość przejazdu obiektu przez sieć transportową (zakręty, ograniczenia prędkości na pewnych liniach). W prezentowanym rozwiązaniu wszystkie elementarne zadania transportowe muszą być realizowane ze stałą prędkością obiektów transportowych na poszczególnych liniach sieci. Nie jest uwzględniane przyspieszenie i opóźnienie związane z ruszaniem i zatrzymaniem się obiektu. Rozwiązanie tego problemu wymaga opracowania odpowiedniego algorytmu, który obliczałby skorygowaną (zmniejszoną) wartość prędkości na danym odcinku, przy której czas pokonania odcinka byłby taki sam, jak dla ruchu, w którym występowałoby przyspieszenie i ewentualnie opóźnienie.

5. PODSUMOWANIE

Zastosowanie modeli symulacyjnych do testowania innowacyjnych rozwiązań w zakresie sterowania systemem produkcyjnym oraz podsystemem transportowym pozwala na sprawdzenie poprawności przyjętych rozwiązań. Jest to szczególnie istotne ze względów bezpieczeństwa. Testowanie niesprawdzonych rozwiązań na obiektach rzeczywistych może doprowadzić do kolizji i uszkodzeń. Zaproponowany w pracy model umożliwiający wyznaczanie dowolnych tras przejazdu obiektów systemu transportowego pozwoli na

wykrywanie zastoju i kolizji spowodowanych ewentualnym i błędami systemu sterującego. Dodatkową zaletą wykorzystania metod symulacyjnych jest niski koszt i łatwość przeprowadzania testów. Ponadto zastosowanie środowiska symulacyjnego Areny w zakresie wizualizacji i animacji symulowanego modelu, w znaczący sposób ułatwi pracę nad testowaniem systemu sterującego.

Pełne przygotowanie modelu do przeprowadzania testów wymaga jeszcze jego rozbudowy. Konieczna jest implementacja algorytmu nadzorującego prędkość, z jaką poruszają się wózki tak, aby czas pokonania trasy przez obiekt rzeczywisty i obiekt w modelu symulacyjnym był taki sam. Uszczegółowienia wymaga rodzaj informacji i format ich przesyłania pomiędzy modelem symulacyjnym a aplikacją sterującą systemem transportowym.

Pracę wykonano w ramach projektu badawczego własnego Nr N N503 214237 pt. "Integracja rozproszonego systemu sterowania produkcją z podsystemem transportu międzyoperacyjnego zbudowanym z autonomicznych wózków mobilnych", finansowanego przez Ministerstwo Nauki i Szkolnictwa Wyższego w latach 2009–2011.

6. BIBLIOGRAFIA

1. Che-Fu Hsueh. A simulation study of a bi-directional load-exchangeable automated guided vehicle system. *Computers & Industrial Engineering* 58 (2010), s. 594–601.
2. B. Cherkassky, A.V. Goldberg, T. Radzik. Shortest Paths Algorithms: Theory and Experimental Evaluation. *Proc. Of 5th Annual ACM-SIAM Symposium on Discrete Algorithms*, Arlington 1994, s. 516–525.
3. R. Fox, A. Garcia, M. Nelson. A Generic Path Planning Strategy for Autonomous Vehicles. The University of Texas – Pan American, Department of Computer Science Technical Report CS-00-25, August, 2000.
4. S. E. Kesen, O. F. Baykoc. Simulation of automated guided vehicle (AGV) systems based on just-in-time (JIT) philosophy in a job-shop environment. *Simulation Modelling Practice and Theory* 15 (2007), s. 272–284
5. L. Qiu, W.-J. Hsu, S.-Y. Huang, H. Wang. Scheduling and routing algorithms for AGVs: a survey. *International Journal of Production Research*. 2002, vol. 40, no. 3, s. 745–760.
6. T. Więk. Laserowy system nawigacji autonomicznej platformy mobilnej na przykładzie urządzenia NAV300. *Pomiary Automatyka Robotyka*, Nr 2, 2011.
7. J. Zajac. *Rozproszone sterowanie zautomatyzowanymi systemami wytwarzania*. Monografia 288, Seria Mechanika. Wydawnictwo Politechniki Krakowskiej, Kraków 2003.
8. J. Zajac. A Deadlock Handling Method for Automated Manufacturing Systems. *CIRP Annals - Manufacturing Technology* 2004, Vol. 53, No. 1, s. 367–370.
9. J. Zajac, G. Chwajło. Koncepcja integracji rozproszonego systemu sterowania produkcją AIM z podsystemem transportu międzyoperacyjnego zbudowanym z autonomicznych robotów mobilnych. *Pomiary Automatyka Robotyka*, Nr 2, 2011.
10. J. Zajac, K. Krupa, A. Słota, T. Więk. Konstrukcja i układ sterowania autonomicznej platformy mobilnej. *Pomiary Automatyka Robotyka*, Nr 2, 2011.
11. Zajac J, Słota A., Chwajło G., *Distributed Manufacturing Control: Models and Software Implementations*. *Management and Production Engineering Review*, Vol. 1., No. 1., May 2010, s. 38–56.