

ANNA ŁATUSZYŃSKA

Uniwersytet Szczeciński

ANALIZA PORÓWNAWCZA OPROGRAMOWANIA DO SYMULACJI WIELOAGENTOWEJ

Wprowadzenie

Symulacja komputerowa jest metodą wykorzystywaną już od wielu lat w różnych dziedzinach wiedzy. Najczęściej traktuje się ją jako efektywne narzędzie do odtwarzania i analizowania szerokiego spektrum złożonych problemów, trudnych do zbadania za pomocą innych metod¹. Symulacja jest metodą opartą na eksperymentach, a jej celem jest uzyskanie wiedzy na temat badanego systemu. Do osiągnięcia tego celu wykorzystywane są programy komputerowe².

Jedną z technik symulacyjnych jest symulacja wieloagentowa. Jej idea wywodzi się z obserwacji emergentnych i adaptacyjnych systemów biologicznych. Początki symulacji wieloagentowej można wywieść od takich dyscyplin, jak systemy adaptacyjne (*complex adaptive systems* – CAS), teoria złożoności (*complexity science*) i teoria systemów (*system science*). W formie zrozumiałej dla komputerów została zaimplementowana po raz pierwszy za pomocą automatów komórkowych, stworzonych niezależnie przez Stanisława Ulama i Johna

¹ A. Champion, *Traffic generation with the SCANeR II simulator: towards a multi-agent architecture*, DSC '99: Proceedings of the first Driving Simulation Conference 1999, s. 311–324.

² M. Łatuszyńska, *Symulacja komputerowa dynamiki systemów*, Wydawnictwo PWSZ, Gorzów Wielkopolski 2008.

von Neumanna w latach czterdziestych XX wieku. Dopiero jednak na początku lat siedemdziesiątych XX wieku zaczęła przybierać formę w jakiej jest znana dzisiaj. Stało się to za sprawą powstania „Gry w życie” (*Game of life*) stworzonej przez Johna Conwaya³. Definicja agentów w obecnym kształcie pojawiła się dopiero na początku lat dziewięćdziesiątych⁴. Od tego czasu rozwój symulacji wieloagentowej znacznie przyspieszył, a w ostatnim dziesięcioleciu zdobyła ona dużą popularność jako metoda badawcza w wielu dziedzinach nauki.

Symulacja wieloagentowa jest oparta na założeniu, że całość systemu jest czymś więcej niż sumą jego poszczególnych części⁵. Modelowanie i symulacja oparte na agentach odzwierciedlają reguły i zadania komponentów systemu w postaci indywidualnych agentów. Każdy agent oddziałuje inaczej, ma jednak udział w ostatecznym wyniku symulacji.

Przy tworzeniu modeli symulacyjnych szczególnie nacisk jest położony na dokładne zamodelowanie pojedynczych agentów. Podczas projektowania agenta, najważniejsze decyzje dotyczą jego zachowania, działań, planów, celów i interakcji z innymi agentami oraz środowiskiem. Dla konstruowania modeli wieloagentowych opracowana została siedmiostopniowa procedura⁶: burza mózgów, teoria, hipotezy, tworzenie diagramu przepływu sterowania dla modelu, kod, analiza i publikacja modelu. Nie jest to jednak jedyna dopuszczalna procedura, inne sugerują, przykładowo, wykorzystanie standardów UML, MESSAGE, PASSI albo Tropos⁷.

Jednym z najważniejszych kroków zmierzających do implementacji modelu wieloagentowego jest wybór oprogramowania, za którego pomocą model będzie tworzony i będzie przeprowadzana symulacja. Można wybierać między trzema wariantami⁸:

- użyciem narzędzia dostosowanego do symulacji, co oznacza, że model będzie adaptowany do specyfiki narzędzia; w kategorii tej mieszczą się wszyst-

³ M. Gardner, *The fantastic combinations of John Conway's New Solitaire Game "Life"*, "Scientific American" 1970, no. 223, s. 120–123.

⁴ J.H. Holland, J.H. Miller, *Artificial adaptive agents in economic theory*, "American Economic Review" 1991, no. 81(2), s. 365–371.

⁵ C.M. Macal, M.J. North, *Managing business complexity – discovering strategic solutions with agent-based modeling and simulation*, Oxford University Press, New York 2007.

⁶ A. Lansdowne, *Traffic simulation using agent-based modeling*, University of the West of England 2006.

⁷ Tamże.

⁸ U. Lotzmann, M. Möhring, *A TRASS-based agent model for traffic simulation*, 22nd EUROPEAN Conference on Modelling and Simulation, ECMS 2008.

- kie programy dedykowane do określonej dziedziny zastosowania;
- użyciem narzędzia symulacyjnego ogólnego przeznaczenia, co pociąga za sobą konieczność dostosowania narzędzia do wymagań modelu;
 - stworzeniem od podstaw narzędzia przeznaczonego do rozwiązania określonego problemu.

Celem artykułu jest analiza narzędzi należących do drugiej grupy – oprogramowania wspomagającego tworzenie modeli ogólnego przeznaczenia. W tej kategorii znajdują się narzędzia oparte na określonym języku oprogramowania, najczęściej obiektowym. Kilka najpopularniejszych zestawiono w tabeli 1.

Tabela 1

Wybrane narzędzia wspomagające tworzenie modeli ogólnego przeznaczenia

Nazwa	Język	Rok	Autorzy
Swarm ⁹	C	1996	Santa Fe Institute
StarLogo ¹⁰	Logo	1996	Mitchel Resnick, Eric Klopfer
Ascape ¹¹	Java	1998	Miles Parker
SeSAm ¹²	Java	1998	Würzburg Universität
NetLogo ¹³	Logo	1999	Uri Wilensky, Center for Connected Learning
Breve ¹⁴	'steve'	1999	Jonathan Klein
Repast ¹⁵	Java	2003	David Sallach, Nick Collier, Tom Howe, Michael North
MASON ¹⁶	Java	2003	George Mason University's Evolutionary Computation Laboratory; GMU Center for Social Complexity

Źródło: opracowanie własne.

W artykule skoncentrowano się na trzech narzędziach wymienionych w tabeli 1. Są to SeSAm, NetLogo i Repast. Zostały one wybrane ze względu

⁹ SWARM, <http://www.swarm.org>, 12.08.2010.

¹⁰ StarLogo, <http://education.mit.edu/starlogo/>, 22.08.2010.

¹¹ Ascape, <http://ascape.sourceforge.net/>, 22.08.2010.

¹² SeSAm, <http://www.simsesam.de/>, 12.08.2010.

¹³ NetLogo, <http://ccl.northwestern.edu/netlogo/>, 12.08.2010.

¹⁴ Breve, <http://www.spiderland.org/>, 12.08.2010.

¹⁵ Repast Simphony, <http://repast.sourceforge.net/docs/reference/SIM/index.html>, 12.08.2010.

¹⁶ MASON, <http://cs.gmu.edu/~eclab/projects/mason/>, 12.08.2010.

na ich popularność i możliwości. SeSAm¹⁷ jest jednym z pierwszych programów opartych na Javie. NetLogo⁶ jest najpopularniejszym rozwiązaniem w zastosowaniach akademickich i naukowych, a Repast Symphony¹⁸ ma największą bibliotekę gotowych rozwiązań i jest często wykorzystywany w komercyjnych rozwiązaniach.

Wyniki analizy porównawczej wymienionych programów pozwolą na określenie, które z badanych narzędzi jest najodpowiedniejsze i najprostsze w obsłudze dla użytkowników niemających doświadczenia w tworzeniu modeli wieloagentowych. Przy użyciu najlepszego z programów zostanie zaprezentowany proces tworzenia i rozwiązywania modelu prostego systemu transportowego.

1. Ogólna charakterystyka wybranych narzędzi

NetLogo¹⁹

Program NetLogo powstał pod koniec lat dziewięćdziesiątych, został stworzony przez Uri Wilenskiego. Do tworzenia modeli wykorzystywany jest język Logo. Nie wymaga on, w przeciwieństwie do innych języków obiektowych, szerokiej wiedzy informatycznej i programistycznej. Eksperymenty symulacyjne są wykonywane w postaci przypominającej proste aplety Javy z wykorzystaniem nieskomplikowanych i efektywnych narzędzi wizualizacji. Z tego powodu NetLogo jest uważane za jeden z wygodniejszych programów, co wpływa na jego dużą popularność. Jako jedyne z badanych narzędzi NetLogo ma wbudowaną klasę agenta, który w programie nosi nazwę *turtle* (żółw). Ułatwia to znacznie tworzenie modeli agentowych również użytkownikom niemającym doświadczenia programistycznego. Z racji tego, że program jest rozwijany już od długiego czasu, ma bardzo rozbudowaną i rzetelną dokumentację, która wydatnie pomaga w zapoznaniu się z jego możliwościami. NetLogo ma również szerokie możliwości wizualizacji efektów symulacji. Dane mogą zostać przedstawione zarówno w postaci graficznej (wykresy, histogramy), jak i liczbowej (monitory). Dane

¹⁷ F. Klügl, F. Puppe, *The multi-agent simulation environment SeSAm*, w: Proceedings des Workshops Simulation in Knowledge-based Systems, ed. H. Kleine Büning, 1998.

¹⁸ N.T. Collier, M.J. North, J. Ozik, E. Tatara, *Visual agent-based model development with Repast Symphony*, Proceedings of the Agent 2007 Conference on Complex Interaction and Social Emergence, 2007.

¹⁹ Dokumentacja NetLogo, <http://ccl.northwestern.edu/netlogo/docs/>, 12.09.2010.

z symulacji mogą również zostać zapisane do pliku za pomocą odpowiedniego polecenia znajdującego się w kodzie. Tworzony model może być zaprezentowany w dwóch lub w trzech wymiarach, a stworzenie intuicyjnego interfejsu do obsługi symulacji jest uproszczone, dzięki możliwości skorzystania z gotowych elementów (przyciski, suwaki, przełączniki), które można umieścić na ekranie metodą „przeciągnij i upuść”. Dzięki niektórym elementom interfejsu, NetLogo zapewnia bardzo uproszczoną zmianę parametrów modelu w czasie symulacji. Dzięki suwakom i przełącznikom można regulować dowolnie, na przykład liczbę agentów i ich właściwości. Wprowadzanie dodatkowych cech agentów i ich metod odbywa się na zasadzie dopisania dodatkowego kodu w języku Logo. Program oferuje także funkcję sprawdzania poprawności kodu, co pozwala uniknąć najczęściej popełnianych błędów składniowych. Wielką zaletą narzędzia jest możliwość jego rozszerzania. W sieci dostępne są pakiety, które pozwalają wzbogacić tworzone modele o dźwięk, moduł GIS, przeprowadzanie symulacji uczestniczących (HubNet), tworzenie filmów i apletów Javy dokumentujących przebieg symulacji oraz przeprowadzanie rozbudowanych eksperymentów symulacyjnych (BehaviorSpace).

SeSAM²⁰

Program SeSAM powstał niemal w tym samym czasie co NetLogo. Według twórców jest to przede wszystkim narzędzie do uproszczonego konstruowania skomplikowanych modeli wieloagentowych. Program jest wykorzystywany głównie w celach edukacyjnych. Tak jak NetLogo, nie wymaga zaawansowanych umiejętności programistycznych. Agenci składają się z ciała, które zawiera zbiór zmiennych stanu i zachowań, jakie są zaimplementowane w formie przypominającej diagramy aktywności w notacji UML (*unified modeling language*). Akcje agentów tworzone są w postaci prostych skryptów, które są inicjowane i kończone na podstawie pewnych reguł. Zachowania agentów i warunki przejścia między stanami są komponowane za pomocą wykorzystania licznych prymitywów (czyli wbudowanych funkcji programu), zaimplementowanych w programie. Pozwala to użytkownikowi na wizualne zaprojektowanie symulacji bez konieczności programowania w określonym języku. Każdy z prymitywów jest opisany w pomocy programu z uwzględnieniem jego działania i wymaganych argumen-

²⁰ M. Fehler, R. Herrler, F. Klügl, *SeSAM: implementation of agent-based simulation using visual programming*, AAMAS 2006.

tów. Jego dokumentacja pozostawia jednak wiele do życzenia w kwestii kompletności. Informacje są rozproszone i nie do końca wyjaśniają wszystkie wątpliwości początkującego użytkownika. Modele przedstawione są zwykle w dwóch wymiarach, choć istnieje również możliwość uzyskania trójwymiarowości. Wymaga to jednak zastosowania rozszerzenia do programu. Analiza przebiegów symulacyjnych może być zaprezentowana na wykresie lub zapisana w tabeli albo pliku. Interfejs modeli jest ustandaryzowany – użytkownik ma jedynie wpływ na wygląd mapy, która reprezentuje środowisko agentów. Dodanie nowych cech i zachowań agentów, podobnie jak w NetLogo, nie sprawia problemów. Jest to kwestia dodania odpowiednich zmiennych i akcji złożonych z prymitywów do definicji agenta. Zmiana parametrów modelu jest jednak nieco bardziej skomplikowana i wymaga stworzenia nowej mapy lub zdefiniowania eksperymentu. W przypadku stwierdzenia błędów w działaniu modelu, program oferuje możliwość uruchomienia symulacji w trybie debuggera, czyli z możliwością krokowego wykonania lub ustawienia punktów kontrolnych.

Tak jak inne programy tego typu, SeSAM oferuje możliwość rozszerzenia swoich funkcjonalności. Dostępne są darmowe pluginy, które umożliwiają między innymi wzbogacenie modeli o dodatkowe opcje komunikacji między agentami, obsługę bazy danych, importowanie danych w plikach z rozszerzeniem .csv, stosowanie modułów GIS oraz zaawansowanych funkcji matematycznych.

Repast Simphony²¹

Wśród wybranych do analizy programów jest to najbardziej rozwinięte narzędzie programistycznie. Repast w całości opiera się na obiektowym języku programowania Java. Mimo pewnych ułatwień dla niezaawansowanych użytkowników (wbudowany prosty model, gotowe interfejsy), do tworzenia modeli wymagana jest przynajmniej podstawowa znajomość składni Javy. W założeniu narzędzie jest głównie przystosowane do symulacji zjawisk z dziedziny nauk społecznych. Agent konstruowany jest w programie za pomocą wbudowanych blozków, które reprezentują akcje i cechy. Każda właściwość agenta jest wyrażona za pomocą pojedynczego elementu, natomiast zachowania są definiowane w formie diagramów przepływu. Zaproponowana metodologia projektowania agentów pozwala na stosunkowo proste rozbudowywanie ich własności i aktyw-

²¹ Dokumentacja Repast Simphony, <http://repast.sourceforge.net/docs/Getting%20Started.pdf>, 12.09.2010.

ności. Dużą zaletą programu jest wykorzystanie dziedziczenia – w modelu mogą wystąpić klasy potomne przejmujące cechy „rodzica”. Podobnie jak w przypadku SeSAm, dokumentacja nie jest zbyt wyczerpująca. Pewną niedogodnością jest również brak biblioteki gotowych modeli, których prześledzenie mogłoby mieć walor edukacyjny.

Strona graficzna programu nie odbiega zbyt od tego, co można było zaobserwować w pozostałych narzędziach. Repast Symphony umożliwia tworzenie dwu- i trójwymiarowych modeli. Do wizualizacji danych dostępne są wykresy i histogramy. Podobnie jak w SeSAMie tworzenie graficznej strony modelu nie obejmuje interfejsu do jego obsługi, a jedynie projekt świata symulacji. Istotną sprawą jest możliwość zmiany parametrów modelu (w tym wartości liczbowych cech agentów) bez konieczności kompilowania całego modelu. Jest to ważne, gdyż proces uruchomienia nowego modelu w formie apletu Javy jest dość czasochłonny.

Repast Symphony jest stworzony głównie jako biblioteka różnorodnych funkcji, które można wykorzystać do tworzenia modeli. Dla zaawansowanych użytkowników stwarza to okazję do dołączania własnych funkcji, dedykowanych do tworzonych projektów, rozszerzających możliwości narzędzia.

2. Założenia i wyniki analizy porównawczej

Analiza porównawcza pakietów do symulacji wieloagentowej została oparta głównie na jakościowych cechach oprogramowania, gdyż wielu elementów, istotnych z punktu widzenia osoby modelującej, nie można bezpośrednio przedstawić liczbowo. Analiza obejmuje następujące aspekty:

- łatwość użycia programu,
- jakość dołączonej dokumentacji,
- możliwości wizualizacji i gromadzenia danych generowanych podczas symulacji,
- możliwości tworzenia interfejsu,
- prostota zmiany parametrów modelu,
- łatwość dodawania nowych właściwości i metod dla agentów,
- możliwość rozszerzania programu o dodatkowe funkcjonalności,
- oryginalne rozwiązania i możliwości każdego programu.

Wszystkie omówione narzędzia zapewniają implementację modelu wieloagentowego, a także wirtualne laboratorium do przeprowadzania eksperymentów symulacyjnych. Mimo analogicznego zastosowania i wielu zbliżonych cech, można jednak wskazać znaczące różnice między opisywanymi pakietami symulacyjnymi. W tabeli 2 zaprezentowano wyniki autorskiej oceny programów NetLogo, SeSAM i Repast Simphony na podstawie wybranych cech, którym przyznano od 1 do 5 punktów.

Tabela 2

Wybrane narzędzia wspomagające tworzenie modeli ogólnego przeznaczenia

Cecha	NetLogo	SeSAM	Repast Simphony
łatwość użycia	4	3	2
dokumentacja	5	1	3
wizualizacja	5	4	4
gromadzenie danych	5	5	4
interfejs	5	4	4
zmiana parametrów	4	1	5
rozbudowa modelu	5	5	5
rozszerzenia	5	5	3
oryginalne rozwiązania	3	3	3
Suma	41	31	33

Źródło: opracowanie własne.

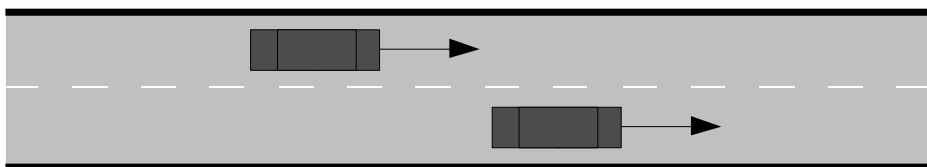
Pod względem łatwości użycia i jakości dokumentacji program NetLogo prezentuje się zdecydowanie najlepiej. W kwestiach wizualizacji i możliwości gromadzenia danych z symulacji, a także w przypadku interfejsu modelu i możliwości jego rozbudowy wyniki są porównywalne. NetLogo i SeSAM wypadają nieco lepiej pod względem liczby dostępnych rozszerzeń zwiększających funkcjonalność programu. Repast Simphony natomiast wyróżnia się pod względem łatwości zmiany parametrów modelu.

Z autorskiego porównania wynika, że najdogodniejszym narzędziem do tworzenia prostych modeli dla niezaawansowanych użytkowników jest NetLogo. Jego możliwości zostały przetestowane przy tworzeniu prostego

modelu odzwierciedlającego ruch pojazdów w jednym kierunku na krótkim odcinku autostrady dwupasmowej.

3. Konstruowanie modelu wieloagentowego w programie NetLogo

Model został zaprojektowany specjalnie na potrzeby niniejszego artykułu. Jego idea została schematycznie przedstawiona na rysunku 1. Model nie obejmuje kolizji na autostradzie i nie przewiduje utrudnień w ruchu spowodowanych pracami remontowymi. Przyjęto w nim, że agentem jest pojazd (dowolnego typu) wraz z kierowcą. Założono, że stanowią oni jedność, która ma zarówno wewnętrzną charakterystykę kierowcy, jak i zewnętrzne właściwości pojazdu²².



Rys. 1. Schematyczne przedstawienie symulowanego systemu

Źródło: opracowanie własne.

Każdy agent – pojazd, ma określone cechy i reguły postępowania, które aktywują odpowiednie metody zaimplementowane dla agenta. Do parametrów agenta możemy zaliczyć: typ (samochód osobowy lub ciężarówka), prędkość początkową (losowaną w momencie utworzenia agenta), prędkość maksymalną i minimalną, a także obszar tolerancji względem innych agentów (to znaczy określenie odległości minimalnych między pojazdami). Każdy agent powinien mieć następujące cechy – metody: zwiększanie prędkości, zmniejszanie prędkości i wyprzedzanie.

Najistotniejszym elementem całego modelu jest stworzenie reguł postępowania dla agentów, a zwłaszcza – określenie reguł interakcji między nimi. Podstawowe zasady obowiązujące wszystkich agentów w modelu można przedstawić w kilku punktach:

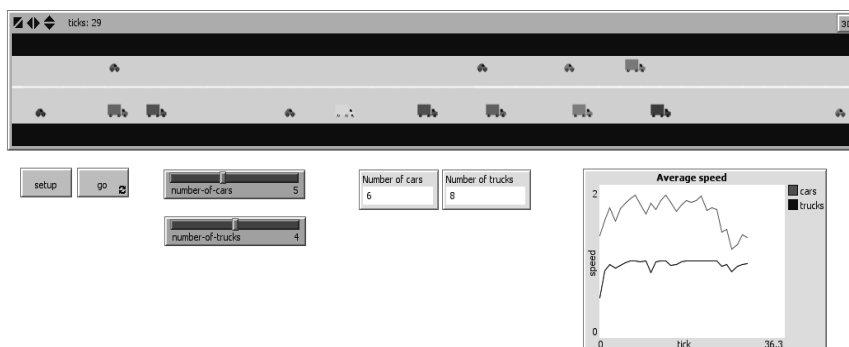
²² D. Helbing, A. Kesting, M. Treiber, *Agents for traffic simulation*, w: *Multi-agent systems: simulation and applications*, ed. A. Uhrmacher, D. Weyns, rozdz. 11, CRC Press 2009, s. 325–356.

1. Agenci pojawiają się na autostradzie z określoną częstotliwością.
2. Prędkość agentów jest losowana zgodnie z przyjętym rozkładem prawdopodobieństwa.
3. Prędkości minimalna i maksymalna oraz typ pojazdu są stałe w ciągu trwania symulacji.
4. Jeżeli odległość między dwoma agentami będzie mniejsza od minimalnej, agent znajdujący się z tyłu zmienia pas na lewy, o ile jest wolny.
5. Jeżeli pas lewy jest zajęty, to agent zwalnia do prędkości agenta poprzedzającego, na tak długo, jak długo pas lewy jest zajęty.
6. Agent porusza się tak długo pasem lewym, dopóki pas prawy jest zajęty. Jeżeli jest wolny, to zmienia pas na prawy.
7. Po osiągnięciu granicy modelowanego obszaru autostrady agent jest usuwany z symulacji.

Budowa dowolnego modelu w NetLogo rozpoczyna się od stworzenia prostego interfejsu użytkownika za pomocą dostępnych w menu elementów i zdefiniowaniu dwóch metod przypisanych do odpowiednich przycisków (*button*). Pierwsza z metod – *setup*, uruchamiana przez przycisk o tej samej nazwie, odpowiada za początkowe wygląd świata symulacji i agentów, druga – *go*, steruje zachowaniem agentów i świata podczas trwania symulacji. W przypadku przykładowego modelu w metodzie *setup* rysowany jest fragment autostrady, na którym losowo umieszczani są agenci. Pasy ruchu są tworzone za pomocą zapisania odpowiedniego kodu w języku Logo. Możliwe jest też importowanie odpowiedniej grafiki. Wśród agentów wyróżnia się dwie „rasy” (*breed*) agentów – pojazdy osobowe i ciężarówki. Początkowa liczba agentów jest określona za pomocą suwaków (*slider*). Jeden suwak określa liczbę samochodów osobowych, drugi – ciężarowych. Każdemu agentowi losowo przydziela się pas ruchu, po którym będzie się przemieszczał, jego kolor i początkową pozycję na autostradzie, a także początkową prędkość. Procedura *go* definiuje pojawianie się w świecie symulacji nowych agentów, zgodnie z rozkładem jednorodnym, a także poruszanie się agentów już obecnych na autostradzie. Z metody *go* wywoływane są pozostałe funkcje odpowiedzialne między innymi za wyprzedzanie samochodów, zmianę pasa, a także zwalnianie i przyspieszanie pojazdu.

Wyniki działania modelu można zaobserwować dzięki dostępnym w NetLogo elementom interfejsu, jakimi są monitory (*monitor*) i wykresy (*plot*), które w omawianym przykładzie pokazują aktualną liczbę ciężarówek i samo-

chodów osobowych na modelowanym odcinku autostrady oraz średnią prędkość pojazdów poszczególnych typów w danej chwili symulacji. Interfejs modelu przedstawiono na rysunku 2.



Rys. 2. Zrzut ekranu przedstawiający model wykonany w programie NetLogo
Źródło: opracowanie własne.

Dodatkowe funkcje zaimplementowane w modelu gwarantują zachowanie odpowiednich odstępów między pojazdami – funkcja ta jest wywoływana zawsze w momencie tworzenia nowego agenta, aby wykluczyć przypadki, gdy dwaj agenci zajmują to samo pole w świecie symulacji. W każdym kroku działania modelu agenci przemieszczają się po autostradzie ze swoją aktualną prędkością. W momencie, gdy dochodzą do końca modelowanego odcinka, zostają usuwani z symulacji. Model jest zaprogramowany na 500 kroków, po których zakończeniu kończy się symulacja. W wyniku jego działania można zaobserwować, jak zmieniała się średnia prędkość pojazdów i ich liczba w określonych przedziałach czasu. Prezentację wyników symulacji można byłoby rozszerzyć o przedstawienie również innych aspektów poruszania się samochodów po autostradzie – w programie NetLogo dodawanie nowych elementów wspierających wizualizację do interfejsu modelu jest bardzo proste, przez co nawet użytkownik bez doświadczenia w dziedzinie symulacji wieloagentowej może przeprowadzać proste eksperymenty.

Podsumowanie

W ostatnich latach pojawiło się wiele programów wspomagających symulację wieloagentową. Programy te, mimo pewnych podobieństw, różnią się znacznie pod względem podejścia do modelowania. W artykule podjęto próbę przybliżenia trzech pakietów symulacyjnych należących do najbardziej popularnych. Analiza porównawcza przeprowadzona według określonych kryteriów, kładących nacisk przede wszystkim na prostotę programu i możliwość jego wykorzystania przez niedoświadczonego użytkownika, pozwoliła na wyróżnienie programu NetLogo. Charakteryzuje się on najlepiej sporządzoną dokumentacją, która pozwala na samodzielną i szybką naukę procedury modelowania oraz na tworzenie własnych modeli na bazie dostępnych w bibliotece. NetLogo ograniczone jest jednak skalą modelu i przede wszystkim sprawdza się przy projektach o małej lub średniej wielkości. Prezentacja przykładowego modelu stworzonego przy użyciu NetLogo pozwoliła na przybliżenie możliwości programu zarówno w zakresie tworzenia agentów i reguł rządzących ich zachowaniem, jak i wizualizacji i gromadzenia danych pozyskiwanych w trakcie symulacji.

Literatura

- Ascape, <http://ascape.sourceforge.net/>, 22.08.2010.
- Breve, <http://www.spiderland.org/>, 12.08.2010.
- Champion A., *Traffic generation with the SCANer II simulator: towards a multi-agent architecture*, DSC '99: Proceedings of the first Driving Simulation Conference 1999.
- Dokumentacja NetLogo, <http://ccl.northwestern.edu/netlogo/docs/>, 12.09.2010.
- Dokumentacja Repast Simphony, <http://repast.sourceforge.net/docs/Getting%20Started.pdf>, 12.09.2010.
- Gardner M., *The fantastic combinations of John Conway's New Solitaire Game "Life"*, "Scientific American" 1970, no. 223.
- Holland J.H., Miller J.H., *Artificial adaptive agents in economic theory*, "American Economic Review" 1991, no. 81(2).
- Helbing D., Kesting A., Treiber M., *Agents for traffic simulation*, w: *Multi-agent systems: simulation and applications*, ed. A. Uhrmacher, D. Weyns, CRC Press, Boca Raton, 2009.
- Fehler M., Herrler R., Klügl F., *SeSAM: implementation of agent-based simulation using visual programming*, AAMAS 2006.

- Klügl F., Puppe F., *The multi-agent simulation environment SeSAM*, w: *Proceedings des workshops simulation in knowledge-based systems*, ed. H. Kleine Büning, Universität Paderborn, 1998.
- Lansdowne A., *Traffic simulation using agent-based modelling*, University of the West of England 2006.
- Lotzmann U., Möhring M., *A TRASS-based agent model for traffic simulation*, 22nd EUROPEAN Conference on Modelling and Simulation, ECMS 2008.
- Łatuszyńska M., *Symulacja komputerowa dynamiki systemów*, Wydawnictwo PWSZ, Gorzów Wielkopolski 2008.
- MASON, <http://cs.gmu.edu/~eclab/projects/mason/>, 12.08.2010.
- Minar N., Burkhart R., Langton C., Askenazi M., *The Swarm simulation system: A toolkit for building multi-agent simulations*, Working Paper 96-06-042, Santa Fe Institute, Santa Fe 1996.
- NetLogo, <http://ccl.northwestern.edu/netlogo/>, 12.08.2010.
- Macal C.M., North M.J., *Managing business complexity – discovering strategic solutions with agent-based modeling and simulation*, Oxford University Press, New York 2007.
- Collier N.T., North M.J., Ozik J., Tataru E., *Visual agent-based model development with repast symphony*, Proceedings of the Agent 2007 Conference on Complex Interaction and Social Emergence, 2007.
- Repast Symphony, <http://repast.sourceforge.net/docs/reference/SIM/index.html>, 12.08.2010.
- SeSAM, <http://www.simsesam.de/>, 12.08.2010.
- StarLogo, <http://education.mit.edu/starlogo/>, 22.08.2010.
- SWARM, <http://www.swarm.org>, 12.08.2010.

COMPARATIVE ANALYSIS OF MULT-AGENT SIMULATION SOFTWARE

Summary

Multi-agent simulation has gained recently a lot of popularity. The models based on agents are used in many fields of science. However, to depict the reality in precise way is very difficult issue. To simplify the development and implementation of multi-agent simulation some specialized tools were created. The article shows the software which focuses on the implementation of already designed model. Three different tools,

which belong to the most popular ones nowadays, were compared. Firstly, short introduction into the multi-agent simulation and its tools is described. In the following sections, on the basis of comparative analysis of them, the most appropriate one for an inexperienced user was chosen. The best software was used to create a simple transportation model. The model shows a sector of a highway and all vehicles moving on it.

Translated by Anna Łatuszyńska