

Jerzy Korczak

Maciej Bac

Krzysztof Drelczuk

Aleksander Fafuła

Uniwersytet Ekonomiczny we Wrocławiu

A-TRADER – DORADCZA PLATFORMA AGENTOWA DLA GRACZY GIEŁDOWYCH

Wprowadzenie

Problem eksploracji danych giełdowych od lat spotyka się z niesłabnącym zainteresowaniem. Opracowano wiele metod, mniej lub bardziej skutecznych, mających na celu analizę rynku i wspomagających proces podejmowania decyzji [Allen 1999; Barbazon 2007; Dempster 2002; Luna 2002; Stanek et al. 2008; Raudys i Zliobaite 2006]. Metody te opierają się na statystyce, ekonomii, sztucznej inteligencji, wiedzy finansowej. W pracy skoncentrowano się jedynie na systemach wieloagentowych i zaproponowano rozwiązanie technologiczne wspomagające podejmowanie decyzji kupna-sprzedaży na rynku papierów wartościowych.

Jednym z pierwszych rozwiązań tego typu jest system wieloagentowy zaproponowany przez K.P. Sycarę, D. Zenga i K. Dechera [Sycara et al. 2002]. System ten umożliwiał użytkownikowi współpracę z wieloma wyspecjalizowanymi agentami, którzy mają dostęp do modeli finansowych i nadzorują na bieżąco pracę systemu, sytuację na rynku, otoczenie oraz realizację priorytetów użytkownika. W ostatnich latach opublikowano wiele interesujących koncepcji systemów wieloagentowych. C. Chiarella, R. Dieci i L. Gardini [Chiarella 2006] opisują system, w którym dwie grupy agentów korzystających z metod analizy fundamentalnej i technicznej modelują dynamikę rynku. Podobne badania przeprowadził F. Westerhoff [Westerhoff 2004]. V. Bohm i J. Wenzelburger [2005] przedstawiają ocenę strategii optymalizacji portfela akcji przez trzech agentów: agenta racjonalnego, agenta z zakłóceniem i agenta analizy technicznej. Spośród

wielu innych prac w zakresie zastosowań systemów wieloagentowych należy również wyróżnić przegląd aktywnych i pasywnych form uczenia agentów działających na rynkach finansowych przez B. LeBarona [2011].

Istnieje kilka standardów opisujących zasady projektowania systemów opierających się na technologiach agentowych. Jednym z nich jest standard stworzony przez organizację FIPA (Foundation for Intelligent Physical Agents). Opisuje on sposoby zarządzania, komunikacji i współpracy pomiędzy agentami. Kolejnym standardem jest MAF (Mobile Agent Facility) zdefiniowany przez Object Management Group – Agent Working Group. Standaryzacją technologii agentowych zajmowały się również The Agents Society oraz Knowledge Sharing Effort. Wyniki prac wyżej wymienionych stowarzyszeń ułatwiają implementację i integrowanie agentów oraz przenoszenie ich pomiędzy istniejącymi systemami. Dostępne implementacje niektórych standardów są w większości implementacjami eksperymentalnymi i niektóre nie są już utrzymywane. Najpopularniejsza z platform JADE jest utrzymywana w dwóch wersjach jako platforma open-source [WWW4] i niezależnie jako platforma komercyjna – JADE 7 [WWW5]. Komercyjna odmiana JADE jest pozycjonowana jako bazowy Framework, dostarczający podstawowe mechanizmy do zastosowań biznesowych. Cechuje się większą szybkością i skalowalnością niż darmowa odmiana. Niestety ta profesjonalna wersja JADE jest darmowa tylko do użytku niekomercyjnego, a jej licencja użytkownika nie dopuszcza modyfikowania kodu źródłowego. Platforma mająca być podstawą do budowy wieloagentowego otwartego systemu do analizy finansowych szeregów czasowych musi być szybka, lekka i łatwa w wykorzystaniu. Ważne są też aspekty bezpieczeństwa i możliwości ulepszenia silnika. Szybkość to nie tylko błyskawiczny protokół komunikacji, ale także natychmiastowe wywołanie reakcji agentów, którzy powinni żadaną informację przetworzyć. Lekkość platformy to niski narzut zużycia zasobów związany z samym procesem odbierania i wysyłania informacji. Niestety wyżej wymienione platformy, mimo iż dostarczają wielu narzędzi i funkcjonalności, które z pewnością byłyby bardzo przydatne, nie spełniają warunków koniecznych, aby posłużyć jako baza dla otwartego systemu wieloagentowego do analizy finansowych szeregów czasowych.

Celem tej publikacji jest przedstawienie projektu i implementacji skalowalnego oraz otwartego systemu wieloagentowego, który dzięki integracji i współpracy agentów opierających się na dowolnych metodach, umożliwiłby jak najlepsze wspomaganie decyzji inwestycyjnych. Istotnymi wymaganiami w realizacji systemu były trafność predykcji, orientacja na ciągłe zdobywanie informacji, niekończące doskonalenie własnej bazy wiedzy oraz zdolność adaptacji do zmieniającego się otoczenia rynku finansowego.

Protoplastą proponowanego rozwiązania wspomagającego podejmowanie decyzji inwestycyjnych na rynku papierów wartościowych był system internet Bourse Expert-Agent Technology (iBE-AT) opracowany w laboratorium LSIT-CNRS w Strasburgu [Korczak i Lipiński 2002; Lipiński 2004]. W systemie iBE-AT zaprojektowano kilka typów agentów: agenty zajmujące się rejestrowaniem i wstępnym przetwarzaniem informacji z rynku papierów wartościowych, agenty szybkiego reagowania bezpośrednio obserwujący rynek w celu natychmiastowego wykrycia anomalii, agenty analizujące zgromadzone dane finansowe i opracowujące decyzje inwestycyjne proponowane użytkownikowi, agenty zarządzające bazą wiedzy i sztucznymi ekspertami finansowymi używanymi do analizy danych finansowych, agenty analizujące dane tekstowe, zwłaszcza informacje prasowe, dotyczące spółek giełdowych, agenty wizualizujące dane finansowe i opracowane decyzje inwestycyjne oraz agenty bezpieczeństwa monitorujące dostęp użytkowników do systemu. Sztuczni eksperci finansowi wskazujący na decyzje kupna-sprzedaży byli wybranymi poprzez algorytm genetyczny podzbiórami reguł analizy technicznej. Każda reguła była funkcją, proponującą jedną z trzech decyzji inwestycyjnych: KUP, SPRZEDAJ lub TRZYMAJ, obliczaną na podstawie ostatnich notowań i wolumenów akcji. Algorytm genetyczny poszukiwał takich podzbiórów reguł, które optymalizowały funkcję celu określającą sprawność finansową danego eksperta w danym okresie.

Innym przykładem zastosowania technologii agentowych przy wspomaganiu podejmowania decyzji inwestycyjnych był system Evolutionary Multi Agent System (EMAS) – [Adamczyk 2007; Korczak 2008], opracowany w Instytucie Informatyki Uniwersytetu Wrocławskiego, składający się z agentów, podobnych funkcjonalnie do agentów iBE-AT analizujących sytuację na rynku i sugerujących decyzje inwestycyjne. W procesie obliczeniowym zbiór agentów podlegał ewolucji, sterowanej przez kilku agentów systemowych, dzięki czemu agenty o słabych umiejętnościach inwestycyjnych były eliminowane i zastępowane agentami lepiej przystosowanymi do warunków rynkowych. System EMAS był oparty na standardach FIPA i zaimplementowany z wykorzystaniem biblioteki JADE, co m.in. umożliwiło uruchomienie systemu na wielu komputerach i prowadzenie obliczeń rozproszonych. Systemy iBE-AT i EMAS zostały przetestowane na rzeczywistych danych pochodzących m.in. z giełdy paryskiej, nowojorskiej i warszawskiej.

Przedstawione powyżej rozwiązania iBE-AT i EMAS nie do końca spełniały oczekiwania użytkowników. Aplikacje cechowało zbyt wolne działanie oraz kosztowne utrzymanie. Ponadto z powodu zastosowania nieaktualnych dzisiaj technologii powstały problemy związane z ich rozwojem czy integrowaniem z aplikacjami innych graczy giełdowych. W celu usprawnienia systemu i usunięcia niedogodności rozpoczęto prace nad platformą A-Trader. W projekcie

A-Trader założono integrację platformy z systemem MetaTrader zasilanym danymi online obejmującymi ticki dowolnych walorów, towarów czy też par walutowych. Tworzone agenty mają dostęp do wybranych danych surowych, jak i danych wstępnie przetworzonych. Opisane w artykule rozwiązanie zostało przetestowane na danych rzeczywistych notowań głównych par walutowych rynku FOREX.

Artykuł został podzielony na trzy zasadnicze części. W pierwszej przedstawiono szkic architektury fizycznej i logicznej proponowanego rozwiązania. Omówiono poszczególne elementy systemu oraz sposób komunikacji pomiędzy nimi. W części drugiej scharakteryzowano problem wstępnej obróbki danych wraz z opisem działania przykładowego agenta odpowiedzialnego za przygotowanie informacji. Przybliżono koncepcję realizacji procesu niekończącego się uczenia. Opisano również nową ideę agenta modelowania zachowań graczy giełdowych i sposób przełożenia zamodelowanych wzorców na sygnały otwarcia i zamknięcia pozycji. W części ostatniej – w podsumowaniu dokonano opisu wyników prowadzonych badań oraz wskazano kierunek ich dalszego rozwoju.

1. Architektura systemu

Prezentowany system ma charakter wieloagentowego rozwiązania wspierającego analizę szeregów czasowych o wysokiej frekwencji, takich jak np. notowania instrumentów finansowych. Podstawowymi jego cechami są otwartość, umożliwiająca integrację i rozwój nowych funkcjonalności systemu oraz zapewnienie odpowiedniej komunikacji między poszczególnymi agentami. Dzięki zorientowaniu na serwisy i przetwarzanie w chmurze rozwiązano problem mocy obliczeniowej. Otwarty i prosty w implementacji protokół komunikacji (SOAP) znacząco ułatwił integrację indywidualnych rozwiązań [WWW1; WWW2; WWW3]. Docelowo protokół ten zostanie zamieniony na własny szybki protokół komunikacji zapewniający wymaganą szybkość przesyłu danych. Popularna w systemach rozproszonych technologia PUSH znacząco przyspieszyła propagację informacji wewnątrz systemu [Agarwal 2011]. Wieloagentowość systemu pozwoliła na personalizację rozwiązania, umożliwiając obserwację dowolnych układów metod użytkownikom końcowym i implementowanym inteligentnym agentom nadzorującym. Agenty oceniające istniejące rozwiązania, przygotowujące zestawy danych uczących oraz metody posiadające zdolność uczenia i dostosowywania się zapewniają ciągłą ewolucję wiedzy opisującej zachowanie rynków finansowych.

W architekturze A-Tradera wyróżniono następujące komponenty:

- a) agenta powiadomień (Notify Agent, NA),
- b) agenta danych historycznych (Historical Data Agent, HDA),
- c) chmurę agentów obliczeniowych (Cloud of Computing Agents, CCA),
- d) agenta komunikacji z rynkiem (Market Communication Agent, MCA),
- e) agenta komunikacji z użytkownikiem (User Communication Agent, UCA),
- f) agenta nadzorującego (Supervisor, S),
- g) bazę danych (System Database, SD).

Agent powiadomień (Notify Agent, NA) zapewnia sprawną komunikację wewnątrz systemu. Jest pośrednikiem w przesyłaniu sygnałów pomiędzy agentami zgodnie z zadeklarowanymi wskazaniem (patrz rys. 1). Każdy agent, którego stan się zmienia powiadamia swojego agenta powiadomień. Agent powiadomień przekazuje informację o zmianie stanu danego agenta wszystkim agentom, którzy są zapisani w rejestrze powiadomień jako klienci/obserwatorzy jego sygnałów. Powiadomienie odbywa się poprzez wywołanie odpowiedniej metody webowej (SOAP) u wszystkich agentów z listy nasłuchujących wskazanych sygnał. Następnie zapisuje informację o zmianie stanu agenta powiadamiającego w bazie danych. Tak pomyślana funkcjonalność agenta powiadomień czyni system elastycznym i skalowalnym, daje możliwość prostego dodawania i usuwania agentów oraz zapewnia uniezależnienie systemu od fizycznego położenia agenta.

Kolejny agent systemu pobiera dane z bazy danych (System Database, SD) i dostarcza je do nowo tworzących lub uczących się agentów, zgodnie z ich potrzebami. Jest nim agent danych historycznych (Historical Data Agent, HDA). Agent ten ma za zadanie zasilenie agentów danymi historycznymi pozwalającymi na ich inicjację bądź wykorzystywani w procesie uczenia się.

Kolejny element systemu stanowi chmura agentów obliczeniowych (Cloud of Computing Agents, CCA), na którą składają się (patrz rys. 1):

- agenty obliczeniowe (Simple Agents Cloud, SAC),
- agenty inteligentne (Intelligent Agents Cloud, IAC),
- agenty własne użytkowników (User Agents Cloud, UAC).

Proste agenty obliczeniowe (Simple Agents Cloud, SAC) to grupa agentów wstępnie przetwarzających dane oraz liczących podstawowe wskaźniki analizy technicznej. Agenty te najczęściej korzystają z nieprzetworzonych danych, rzadziej z danych już przekształconych przez inne agenty CCA. Wykonują one predefiniowane operacje o niskiej złożoności pamięciowej i obliczeniowej. Umożliwia to szybkie uzyskanie wyników i przekazanie ich dalej do użytku agentów o bardziej zaawansowanej logice. Wynikiem działania prostych agentów obliczeniowych jest stały zestaw danych opisujących rynek, które są wykorzystane

przez pozostałych agentów. Z punktu widzenia implementacji rozwiązania, SAC mogą być integralną częścią agenta powiadomień, co pozwoli zaoszczędzić czas przesyłania danych i grupowania ich przy wysłaniu do agentów docelowych.

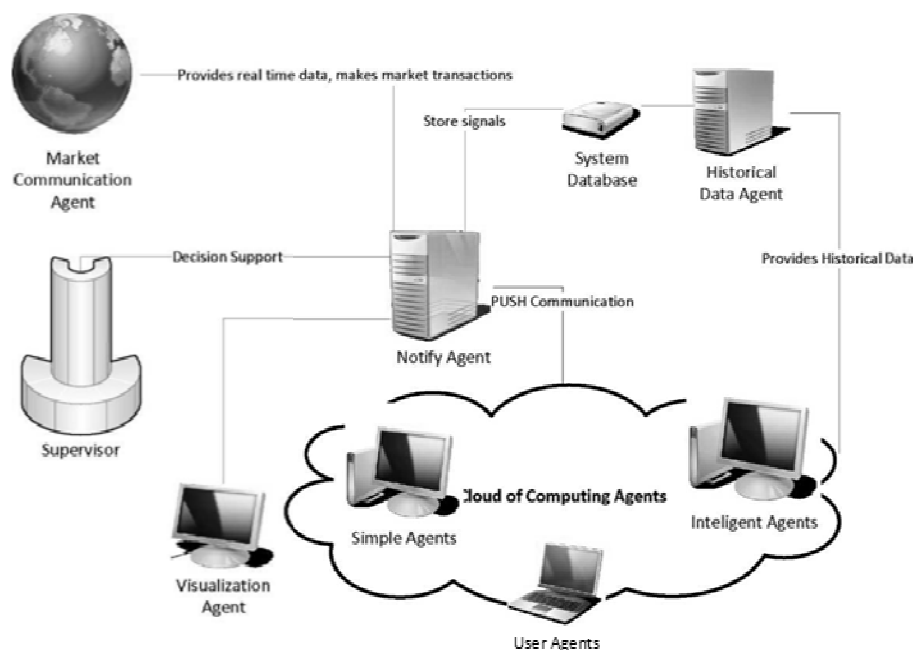
Agenty posiadające własną bazę wiedzy, mogące się uczyć, zmieniać swoje parametry oraz zmieniać swój stan wewnętrzny tworzą kolejny element chmury agentów, zwanych agentami inteligentnymi (Intelligent Agents Cloud, IAC). Możliwość obserwacji wyników dostarczanych przez innych agentów, przetwarzania ich „zapamiętywania” oraz uczenia się powoduje, że agenty te mogą nieustannie zmieniać swoją logikę działania i poprawiać skuteczność. Do tej grupy agentów zalicza się wszystkie rozwiązania opierające się na sztucznej inteligencji (algorytmy genetyczne, sieci neuronowe, systemy ekspertowe itd.), agenty analizujące komunikaty tekstowe, agenty obserwacji zachowań użytkownika. Wśród tych agentów należy wyróżnić: agenty wstępnej transformacji szeregów finansowych, agenty modelowania zachowań graczy giełdowych, agenty opierające się na niekończącym się uczeniu.

Agenty własne użytkowników (User Agents Cloud, UAC) to z kolei chmura agentów, w której znajdują się agenty utworzone przez użytkowników wewnętrznych. Wydzielenie tej części systemu otwiera możliwość integracji własnych agentów z całym rozwiązaniem bez konieczności instalacji agenta na serwerach zespołu badawczego. Takie podejście zapewnia, że z jednej strony badacz z poza zespołu nie musi zdradzać tajemnicy działania swojego agenta, a mimo to odbiera i wysyła sygnały. Z drugiej strony zewnętrzny agent nie zagraża stabilności działania charakteryzowanego systemu. Dzięki temu zewnętrzni użytkownicy mogą swobodnie dołączać do systemu swoich agentów, a dołączone agenty mogą korzystać ze wszystkich informacji/sygnałów dostarczanych przez system. Agenty inteligentne mogą otrzymywać i interpretować sygnały dostarczane przez UAC. Po uzyskaniu odpowiedniej stabilności agenty własne mogą być przeniesione do chmury agentów inteligentnych.

Komunikacja systemu z otoczeniem zewnętrznym jest zapewniana dzięki agentom komunikacji z rynkiem (Market Communication Agents, MCA). Z jednej strony agenty te dostarczają wiadomości z rynków finansowych oraz notowania dostępnych walorów. Z drugiej strony są odpowiedzialne za realizację zleceń otwarcia bądź zamknięcia pozycji. W tej grupie znajdują się agenty analizujące informacje ze spółek oraz przeglądające treści portali piszących o rynkach finansowych. Mogą one m.in. dostarczyć informacji na temat nastrojów inwestorów/spekulantów poprzez analizę blogów ekspertów finansowych lub przeglądanie odpowiednich wpisów na portalach społecznościowych.

Szybka i łatwa wizualizacja wyników pracy agenta, czy też całego systemu, jest ważnym aspektem w weryfikacji poprawności jego działania. W systemie jest ona możliwa dzięki agentom komunikacji z użytkownikiem (User Communication Agents, UCA). Narzędzia wizualizacyjne pozwalają na szybką analizę działania agenta, co może znacząco przyczynić się do ułatwienia prac nad poprawą jego skuteczności. Jest to również sposób przedstawienia wyników platformy użytkownikom końcowym. Agent komunikacji pozwala użytkownikowi na przekazanie własnych sugestii agentom inteligentnym. Umożliwia zmianę parametrów wybranego agenta, czy też sugestie dla agenta nadzorującego, dotyczącą mechanizmów i tego w jakim stopniu mają one wpływać na decyzje inwestycyjne. UCA pozwala użytkownikowi na przygotowanie i analizę zestawów uczących dla agentów inteligentnych. Prezentacja sygnałów dostarczonych przez agentów charakteryzowanego systemu nie wymaga implementacji własnych narzędzi, ale jest możliwa dzięki integracji agentów z oprogramowaniem dostarczonym przez brokerów.

System wieloagentowy charakteryzuje się tym, że agenty mogą generować niezależne decyzje. Mogą to być decyzje zgodne lub zupełnie przeciwstawne. Takie wykluczające się decyzje to np. sugestia otwarcia i zamknięcia pozycji wygenerowane przez dwóch niezależnych agentów w tym samym czasie. Następuje wtedy konflikt pomiędzy agentami. Rozwiązywaniem konfliktów zajmują się agenty nadzorujące (Supervisor, S), które obserwują decyzje wszystkich agentów obliczeniowych i inteligentnych, oceniają ich skuteczność w inwestowaniu oraz ryzyko. Są odpowiedzialne za rozwiązanie konfliktu. Na podstawie zgromadzonych informacji decydują, które agenty są brane pod uwagę w podejmowaniu decyzji inwestycyjnej, a których podpowiedź jest pomijana. Do rozwiązywania konfliktów mogą być zastosowane drzewa decyzyjne lub metody negocjacji [Korcza 2006]. Skuteczne może też okazać zaimplementowanie agenta supervisora opierającego się na metodzie konsensusu [Hernes 2011]. Należy zaznaczyć, że decyzja podjęta przy pomocy metody konsensusu uwzględnia sygnały wszystkich agentów, co zmniejsza ilość błędnych decyzji. Agent nadzorujący poza podejmowaniem ostatecznej decyzji reguluje wielkość pozycji i czas ich utrzymywania. Odpowiada za bezpieczeństwo inwestycyjne całego systemu i w razie wystąpienia na rynku sytuacji wyjątkowych lub nieprzewidywanych przez system, zaprzestaje inwestycji.



Rys. 1. Architektura systemu A-Trader

Zaproponowana architektura jest zgodna ze standardem FIPA. Wszystkie wymagane komponenty i usługi zapewnia agent powiadomień. Agent powiadomień jest systemem zarządzania agentami (Agent Management System). Przechowuje informacje o dostępnych agentach, o ich lokalizacji (White Pages) i o udostępnianych usługach (Yellow Pages), pełni rolę usługi katalogowej (Directory Facilitator). Pośredniczy w przekazywaniu wiadomości pomiędzy poszczególnymi agentami (Message Transport). Specyfika rozwiązywanego problemu (generowanie sygnałów otwarcia i zamknięcia pozycji), ze względu na dynamicznie zmieniające się środowisko, wymaga bardzo wysokiej sprawności systemu. Zaawansowana komunikacja pomiędzy agentami występuje jedynie w procesie uczenia całego systemu lub jego poszczególnych agentów. Agent rozpoczynając pracę w trybie online uczy się bądź korzysta ze swojej bazy wiedzy i uzupełnia potrzebne dane historyczne. Gdy jest już przygotowany do działania, jego praca sprowadza się do analizy nadsyłanych danych od agentów powiadomień, przy wykorzystaniu wcześniej zgromadzonej wiedzy i przekazywaniu swojej decyzji do swojego agenta powiadomień. Kiedy trzeba natychmiast podjąć decyzję nie występuje proces zaawansowanej negocjacji międzyagentowej, takiej jak ustanowienie połączenia czy też uzgodnienie protokołu komunikacji.

W celu ułatwienia implementacji każdy agent jest uruchamiany w obrębie swojego kontenera izolującego od środowiska i enkapsulującego komunikację z centralną częścią systemu: agentem powiadomień. Wiele kontenerów może być uruchomionych na jednej maszynie. Priorytety przy projektowaniu kontenera były następujące:

- maksymalne uproszczenie procesu tworzenia agenta i jego integracji z systemem, praktycznie przezroczystej dla użytkownika,
- maksymalne uproszczenie uruchomienia już stworzonego agenta oraz włączenie go do chmury.

Celem wyizolowania agentów i przeniesienia ich do chmury jest z jednej strony zapewnienie asynchronicznej współpracy, z drugiej umożliwienie wykonywania wyspecjalizowanych działań na dedykowanych do tego środowiskach, np. algorytmy obliczeniowe, które mogą być wykonywane współbieżnie na komputerach wyposażonych w wieloprocessorowe karty graficzne NVIDIA. Agenty wizualizacyjne mogą być też natomiast uruchamiane na urządzeniach mobilnych ułatwiających dostęp użytkownikom do wyników obliczeń chmury.

W ramach prototypu został zaimplementowany kontener na platformie .NET, co umożliwia uruchomienie agentów na komputerach z systemami rodziny Windows i Unix. W kolejnych etapach planuje się utworzenie kontenerów dla agentów JAVA oraz umożliwiających integrację z popularnymi środowiskami, takimi jak np. MATLAB.

2. Charakterystyka funkcjonalna wybranych agentów

2.1. Agenty wstępnego przetwarzania danych

Wstępna obróbka danych zgodnie z metodologią CRISP-DM (Cross-Industry Standard Process for Data Mining) – [Chapman et al. 2000] opracowana przez analityków Daimler-Chrysler, SPSS i NCR, jest zaliczana do drugiego i trzeciego etapu procesu eksploracji danych, czyli „zrozumienia danych” i „przygotowania danych”.

Większość danych przechowywanych w bazach danych jest nieobrobiona, niekompletna i zaszumiona. Przykładowo baza może zawierać:

- wartości, które są przestarzałe lub zbędne,
- notowania z brakującymi wartościami,
- notowania nietypowe (ang. *outliers*),
- dane w nieodpowiednim formacie,
- wielowymiarowe zmienne zależne itp.

Nadrzędnym celem procesu jest minimalizacja danych nieistotnych, które dostają się do modelu i powodują często błędne wyniki.

Bardzo dużym problemem przy wstępnej analizie i obróbce szeregów finansowych o dużej frekwencji jest minimalizacja czasu przygotowania danych. Należy zaznaczyć, że algorytmy te przygotowują dane dla modeli, których czas predykcji musi też być bardzo krótki. Projektanci modeli eksploracji danych bardzo często, balansując pomiędzy tymi dwoma wytycznymi, muszą godzić się na kompromisy odnośnie do jakości danych, aby udostępnić je w akceptowalnym czasie reakcji systemu.

Agenty odpowiedzialne za wstępne przetwarzanie danych, w prezentowanym systemie, są elementem chmury agentów obliczeniowych. Istotnym kryterium postawionym przed takimi agentami jest szybkość przetwarzania. Wyekstrahowanie takiej grupy agentów, a co z tym idzie oddelegowanie zadań związanych z wstępnym przetwarzaniem danych do wyspecjalizowanych agentów, stwarza możliwość efektywnego zarządzania procesem wstępnej analizy danych, m.in. poprzez równoległą realizację algorytmów w celu uzyskania danych o wysokiej jakości w akceptowalnym czasie.

2.2. Agenty niekończącego się uczenia

Agenty odpowiedzialne za niekończące się uczenie są elementem charakteryzowanej wcześniej chmury agentów, a konkretnie grupy agentów inteligentnych. Koncepcja „never ending learning” jest w tym przypadku realizowana na podstawie czterech grup agentów: agentów grupujących, agentów trenerów, agentów oceniających i grupy agentów, którzy mają być podmiotem działań algorytmu.

Zaproponowana metoda niekończącego się uczenia opiera się na następujących założeniach:

- a) metoda nie ogranicza użytkownika w wyborze instrumentów finansowych. W projekcie platformy wykorzystuje się dane rzeczywiste z rynku walutowego, jednak algorytm będzie sprawnie działać na wszystkich innych szeregach czasowych o wysokiej frekwencji;
- b) liczba agentów jest nieograniczona;
- c) algorytm agenta może opierać się na dowolnym algorytmie, który można dostosowywać do zmieniających się warunków rynku finansowego poprzez zmianę parametrów, do których można m.in. zaliczyć sieci neuronowe [Bac 2010], algorytmy ewolucyjne [Bac i Kwaśnicka 2009], reguły asocjacyjne, czy systemy ekspertowe [Dymova et al. 2010]. Algorytmy parametryzowane to też np. funkcje analizy technicznej (np. RSI, MACD, średnia krocząca

z parametrem wielkości okna czasowego) lub też inne algorytmy niezmienniejące swojego stanu wiedzy, lecz dające się modyfikować poprzez parametry wejściowe.

Na grupę agentów niekończącego się uczenia składają się: agent grupujący, agent trener, agenty oceniające i grupa agentów samouczących. Agent grupujący (asocjacji) wyróżnia agentów, którzy skutecznie ze sobą współdziałają. Część agentów będzie miała wyższą skuteczność podejmowanych decyzji przy aktywności (konkretnej wartości sygnału wyjściowego) innych agentów. W ten sposób agent grupujący będzie w stanie wyróżnić agentów, którzy skutecznie działają przy dużej lub małej zmienności albo też przy rynku wysprzedanym lub wykupionym. Ponadto agent grupujący dzieli okresy na obserwowanych szeregach czasowych na klastry, opisując je przy pomocy konkretnych wartości sygnałów wyjściowych poszczególnych agentów. Następnie agent trener tworzy na podstawie klastrów nowe instancje agentów znacznie skuteczniejszych w wyznaczonych warunkach. Takie działanie powoduje wytworzenie się bardzo wyspecjalizowanych agentów, którzy mają ponadprzeciętną skuteczność inwestycyjną dla konkretnych sytuacji/faz, w których znajdzie się rynek. Agenty oceniające weryfikują skuteczność agentów i wskazują przyczynę niskiej efektywności (zbyt duża liczba transakcji, przedwczesne lub zbyt długie przetrzymywanie pozycji, nieprawidłowy TP – Take Profit lub SL – Stop Loss). Przyspiesza to znacząco specjalizację agentów posiadających zdolności autoadaptacyjne. Powstawanie nowych agentów wykrywających zupełnie nowe fazy rynku pozwoli agentowi grupującemu na stworzenie nowych klastrów, z kolei zgodnie z koncepcją „never ending learning” umożliwi ewolucję systemu.

2.3. Agent Modelowania Zachowań Graczy

Jednym z elementów systemu A-Trader jest nowatorski komponent – system eksploracji zachowań inwestorów z danych giełdowych. Podejście to jest próbą formalnego ujęcia elementów finansów behawioralnych, a w szczególności automatycznego wykrywania błędów poznawczych.

Algorytmy agenta są efektem kilkuletnich badań, prowadzonych przy użyciu specjalistycznego symulatora giełdowego Etoile [Fafuła 2010]. Agent jest wciąż w fazie doskonalenia i rozbudowy. Obecnie możliwe do rozpoznania są cztery sytuacje:

- a) paradoks hazardzisty (podczas sprzedaży i kupna) – wiara w zmianę trendu wynikająca z błędnego obliczania prawdopodobieństw, w szczególności szacowania regresji do średniej [Tversky 1974],

b) efekt *hot hand* (podczas kupna) – [Sundali i Croson 2006] i *cold hand* (podczas sprzedaży) – w uproszczeniu: przeświadczenie o kontynuacji, np. „dobre spółki zawsze zarabiają”, „słabe firmy nigdy się nie wybiją”.

Podczas konstrukcji tego złożonego, wykorzystującego liczne technologie, platformy i algorytmy, agenta przeprowadzono wiele eksperymentów z zakresu psychologii inwestowania. Decyzje i stany emocjonalne przeszło stu osób zostały naniesione maszynowo na przestrzeń opisywaną przez analizę techniczną. Łącznie zebrano kilka tysięcy sygnałów kupna i sprzedaży. Doświadczenia odbywały się w różnych fazach rynku i na kilkuset spółkach. Następnie, opierając się na pozyskanych danych, a przy wykorzystaniu indukcyjnych drzew decyzyjnych, zostały wygenerowane reguły decyzyjne. Ostatecznie agent modelowania zachowań korzysta z wybranych reguł, wysyłając sygnały kupna i sprzedaży. Przykładowo, w działaniu, Agent Modelowania Zachowań w wariacie Gambler's Fallacy sygnalizuje zwiększone prawdopodobieństwo pobudzenia się kontrarian.

Dotychczas system ten był testowany autonomicznie na giełdzie NASDAQ, gdzie zaprezentował obiecujące wyniki. Sprawdza się najlepiej na notowaniach o dużej płynności. Zarówno do obliczeń, jak i mapowania sytuacji agent korzysta z danych dziennych.

Podsumowanie

Pierwsze próby zaimplementowanego środowiska wieloagentowego okazały się bardzo zachęcające. Agent nadzorujący zmniejszył ryzyko inwestycyjne ograniczając samodzielne działania bardziej ryzykownych agentów na rzecz wspólnych decyzji całego środowiska. Współpracujące agenty dawały częściej rentowne decyzje i znacznie wcześniej zamykały pozycje przynoszące straty.

Przyjęty do komunikacji protokół SOAP pozwolił na szybką implementację pojemników na agenty w środowisku C#. Autorzy w dalszych pracach planują implementacje kontenerów w języku Java, PHP oraz środowiska dla rozwiązań tworzonych w oprogramowaniu Matlab. Zastosowanie kontenerów umożliwi zamianę protokołu komunikacji SOAP na standard binarny gwarantujący szybszą komunikację.

Platforma pozwala na integrację nowych rozwiązań i metod inteligentnych działających w obszarze analizy szeregów finansowych. Pozwala na wykorzystanie sygnałów generowanych przez już zaimplementowanych agentów, dzięki czemu można testować skuteczność własnych metod w powiązaniu z algorytmami zaimplementowanymi dotychczas. Budując własne agenty, można korzystać z informacji agentów oceniających w celu poprawiania własnych rozwiązań.

Opisany system wieloagentowy ułatwia testowanie i walidację nowych algorytmów poprzez dostarczenie podstawowych funkcjonalności oraz danych. Umożliwia to koncentrację prac na budowaniu kolejnych agentów, nie dbając o podstawowe mechanizmy dostarczania danych oraz komunikacji.

Innowacyjnym pomysłem wykorzystania infrastruktury jest stworzenie agentów symulujących zachowania uczestników handlu na rynku. Stworzenie przekroju poprzez „typy” inwestorów, uwzględnienie proporcji pomiędzy liczebnością inwestorów każdego typu oraz uwzględnienie zasobów kapitałowych poszczególnych kategorii inwestorów umożliwi przewidywanie przepływów kapitału. Ideę tę można spróbować zastosować do zmian alokacji kapitału na obserwowanym walorze oraz zmian pomiędzy walorami na wybranym rynku i alternatywnymi możliwościami inwestycyjnymi (papiery wartościowe, obligacje, towary, waluty).

Platforma A-Trader jest w fazie testowania i rozbudowy. Ilość i zakres wykorzystywanych metod jest przez cały czas rozszerzany. Powstają nowe agenty opierające się na najnowszych metodach sztucznej inteligencji. Na podstawie przeprowadzanych badań są rozwijane rozwiązania działające na różnych rynkach i wykorzystujące dostępne algorytmy. Jest to otwarta platforma do testowania modeli wykorzystywanych w pracach naukowych zespołu badawczego oraz pracach dyplomowych studentów.

Literatura

- Adamczyk I. (2007): *Ewolucyjny system wieloagentowy*. Praca magisterska, Uniwersytet Wrocławski, Wrocław 2007.
- Agarwal S. (2011): *Toward a Push Scalable Global Internet*. Proc. Global Internet Symposium. IEEE Infocom, Shanghai.
- Allen F., Karjalainen R. (1999): *Using Genetic Algorithms to Find Technical Trading Rules*. „Journal of Financial Economic”, No. 51.
- Bac M. (2010): *Self Organizing Map (SOM) Network Application Support for Short-term Investment Decisions*. W: *Data Mining and Business Intelligence*. Red. J. Korczak. Wydawnictwo UE, Wrocław.
- Bac M., Kwaśnicka H. (2009): *Możliwości zastosowania algorytmów genetycznych w systemach informacyjnych wspomagających proces podejmowania decyzji gracza giełdowego*. W: *Inżynieria i systemy ekspertowe*. Red. A. Grzech, K. Juszczyszyn, H. Kwaśnicka. Akademicka Oficyna Wydawnicza Exit, Warszawa 2009.
- Barbazon A., O'Neill M. (2007): *Biologically Inspired Algorithms for Financial Modeling*. Springer, 2005.
- Bohm V., Wenzelburger J. (2005): *On the Performance of Efficient Portfolios*. „Journal of Economic Dynamics and Control” April, Vol. 29, Iss. 4.

- Chapman P., Clinton J., Kerber R., Khabaza T., Reinart T., Shaerer C., Wirth R. (2000): *CRISP-DM Step-by-step Data Mining Guide*, <http://www.crisp-dm.org>.
- Chiarella C., Dieci P., Gardini L. (2006): *Asset Price and Wealth Dynamics in a Financial Market with Heterogeneous Agents*. „Journal of Economic Dynamics and Control”, Vol. 30.
- Dempster M., Jones C. (2001): *A Real Time Adaptive Trading System using Genetic Programming*. „Quantitative Finance”, No. 1.
- Dymova L., Sevastianov P., Bartosiewicz P. (2010): *A New Approach to the Rule-based Evidential Reasoning: Stock Trading Expert System Application*. „Expert Systems with Applications”, Vol. 37, Iss. 8.
- Fafuła A. (2010): *A Prototype of Platform for Data-driven Approach to Detection of Cognitive Biases*. W: *Data Mining and Business Intelligence*. Red. J. Korczak. Wydawnictwo UE, Wrocław.
- Hernes M. (2011): *Metody konsensusu w rozwiązywaniu konfliktów wiedzy w wieloagentowym systemie wspomaganie decyzji*. Praca doktorska. Wydawnictwo UE, Wrocław.
- Korczak I., Lipiński P. (2008): *Systemy agentowe we wspomaganie decyzji na rynku papierów wartościowych*. W: *Rozwój informatycznych systemów wieloagentowych w środowiskach społeczno-gospodarczych*. Red. S. Stanek et al. Placet, Warszawa.
- Korczak J., Lipiński P. (2002): *Design of Stock Trading System Based on Intelligent Agents*. Internat. Conf. ACs'2002, Międzyzdroje.
- Korczak J., Lipiński P. (2006): *Technology of Intelligent Agents used in Financial Data Analysis*. Proceedings of 5th Ogólnopolska Konferencja Naukowa Nowoczesne Technologie Informacyjne w Zarządzaniu, NTIZ2006. Wydawnictwo AE, Wrocław.
- Le Baron B. (2011): *Active and Passive Learning in Agent-based Financial Markets*. „Eastern Economic Journal”, Vol. 37.
- Lipiński P. (2004): *Evolutionary Data-Mining Methods in Discovering Stock Market Expertise from Financial Series*. PhD thesis, Université Louis Pasteur, Strasbourg 2004.
- Luna F., Perrone A., eds (2002): *Agent-based Methods in Economics and Finance: Simulations in Swarms*. Springer.
- Raudys S., Zliobaite I. (2006): *The Multi-Agent System for Prediction of Financial Time Series*. Proc. ICAISC, LNAI4029.
- Stanek S., Sroka H., Paprzycki M., Gannzha M. (2008): *Rozwój informatycznych systemów wieloagentowych w środowiskach społeczno-gospodarczych*. Placet, Warszawa.
- Sundali J., Croson P. (2006): *Biases in Casino Betting: The Hot Hand and the Gambler's Fallacy*. „Judgment and Decision Making”, Vol. 1, No. 1.
- Sycara K.P., Decker K., Zeng D. (2002): *Intelligent Agents in Portfolio Management, Agent Technology: Foundations, Applications, and Markets*. Eds. N. Jennings, M. Wooldridge. Springer, Heidelberg.
- Tversky A., Kahneman D. (1974): *Judgment under Uncertainty: Heuristics and Biases*. „Science” 185 (4157).
- Westerhoff F.H. (2004): *Multiasset Market Dynamics. Macroeconomic*. „Dynamics”, No. 8.

[WWW1] <http://www.w3.org/TP/2003/WD-soapl2-mtom-20030721/>

[WWW2] <http://www.w3.org/TP/soap/>

[WWW3] <http://www.xml.com/pub/a/2003/02/26/binaryxml.html>

[WWW4] <http://iade.tilab.com/>

[WWW5] <http://www.jade.co.nz/jade/index.htm>

A-TRADER – ADVISORY MULTI-AGENT PLATFORM FOR STOCK TRADERS

Summary

The authors of this paper present the architecture of a multi-agent system which supports investment decisions on the stock market. The individual components of the system, the manner of communication between them, the mechanism of assessing the individual agents are discussed. New methods of pre-processing financial series, the concept of never-ending learning, the behavioural model of stock exchange traders and the manners of translating the modelled patterns into the open and close position signals are described. The results of the research are described and the directions of the further development of the platform are provided in the conclusion.