

11.

Modelowy agentowy system e-commerce

Maria Ganzha, Maciej Gawinecki,
Paweł Kobzdej, Marcin Paprzycki

11.1. Wprowadzenie

Pojęcie *E-Commerce* (*electronic commerce*) można definiować rozmaicie. Jedną z definicji przedstawiła Światowa Organizacja Handlu, opisując e-commerce jako produkcję, reklamę, sprzedaż i dystrybucję produktów przez sieci teleinformatyczne [41]. E-commerce działa głównie w technologii WWW, jednakże wymaga również innych usług (np. usługi transportowe w przypadku produktów, które należy dostarczyć do klienta) [12].

Badania podsumowane w pracach [27, 28] pokazały, że technologia agentowa ma szansę podwyższyć rentowność przedsiębiorstw wirtualnych. Dzieje się tak między innymi dzięki możliwości taniego zwiększenia liczby negocjowanych i zawieranych transakcji oraz skróceniu czasu zakupu. Natomiast klient może zyskać na zastosowaniu technologii agentowej przez uzyskanie ceny towaru zbliżonej do jego rzeczywistej wartości rynkowej. Jest to rezultatem:

1. dostępu agentów do pełniejszej informacji o cenie towaru na rynku oraz
2. możliwości uczestniczenia w negocjacjach cenowych i „testowaniu hipotez” na temat tego, jaką cenę można uzyskać na rynku.

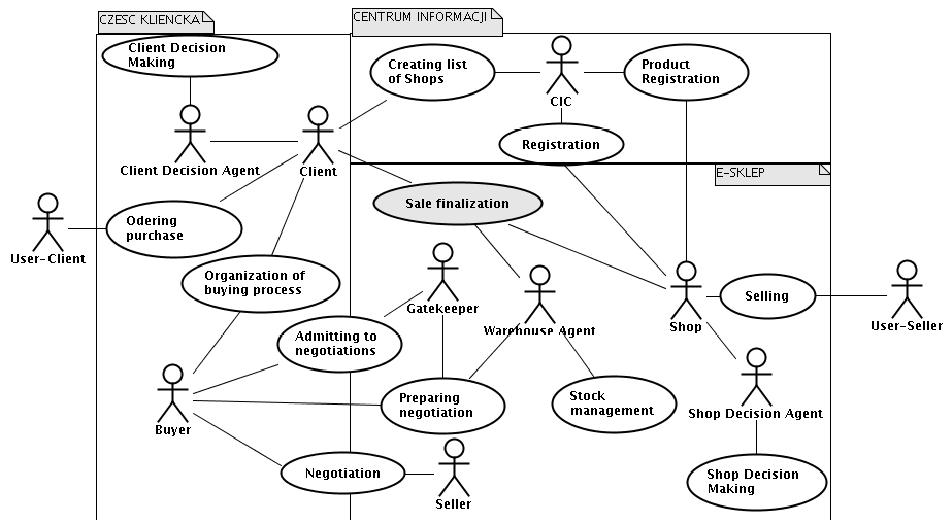
11.2. Opis modelowego systemu e-commerce

Przedstawimy teraz architekturę proponowanego systemu e-commerce modelującego procesy zakupu/sprzedaży z użyciem agentów. Według [40] proces zakupu/sprzedaży obejmuje następujące fazy:

- *faza wstępna* – zawiera działania typu: badanie zapotrzebowania, zamawianie towaru, itp.;
- *negocjacje* – w czasie których uczestnicy negocjują cenę zgodnie z obowiązującymi regułami, używając do tego celu swoich prywatnych strategii;
- *finalizacja transakcji* – potwierdzenie zamówienia, dostawa towaru i odbiór należności;
- *faza końcowa* – zbieranie informacji z procesu i wyciąganie na tej podstawie wniosków na przyszłość.

Dwie pierwsze fazy – wstępna i negocjacje – nadają się szczególnie do zastosowania w nich technologii agentowej, dlatego im właśnie poświęcono znaczną część niniejszego rozdziału. Jednakże zwracać będziemy również uwagę na to, jaka informacja generowana w systemie może zostać wykorzystana do wyciągania wniosków na przyszłość.

Schemat przypadków użycia systemu, wraz z występującymi w nim agentami i interakcjami między nimi, przedstawiony został na rysunku 11.1.



Rysunek 11.1. Przypadki użycia w prezentowanym systemie

Źródło: opracowanie własne.

Możemy wyodrębnić trzy główne części systemu:

1. *centrum informacji*, gdzie przechowywane są informacje o tym, który sklep sprzedaje jakie produkty;
2. *część kliencką*, w skład której wchodzi agenci reprezentujący interesy kupującego;
3. *e-sklep* – reprezentowany przez agentów realizujących zadania sprzedawcy.

Załóżmy teraz, że system działa już przez jakiś czas. Zostały więc zainicjowane funkcje wszystkich jego komponentów (np. e-sklepy zarejestrowały swoje towary w centrum informacji). Jak odbywa się obsługa kolejnego zamówienia użytkownika? Klient przekazuje swojemu przedstawicielowi – Client Agent (CA) – informację o tym, co chciałby kupić. Na podstawie tych danych CA zwraca się do centrum informacji z prośbą o listę sprzedawców danego towaru. Po otrzymaniu listy sklepów, CA wybiera spośród nich te, które są godne zaufania i umieszcza w nich swoich przedstawicieli (Buyer Agent; BA) w celu uczestnictwa w negocjacjach. Decyzję gdzie kupić potrzebny towar (lub, że przy zadanych parametrach zakup jest niemożliwy) CA podejmuje na podstawie raportów o wynikach tych negocjacji nadchodzących od BA.

Z punktu widzenia sprzedającego system wygląda następująco. Sprzedawca jest reprezentowany przez agenta Shop Agent (SA). Ustala on listę towarów oraz rejestruje swój sklep i towary w centrum informacji (CIC) i czeka na klientów. W procesie sprzedaży agenta SA wspiera Gatekeeper Agent (GA). Jest to agent, który wpuszcza do sklepu tylko tych kupców (BA), którzy są godni zaufania. Następnie przekazuje im niezbędne informacje o mechanizmie sprzedaży danego produktu; po czym, gdy tylko kupiec poinformuje, że jest gotów przystąpić do negocjacji, dodaje go do puli kupców zainteresowanych tymże towarem. Gdy proces gromadzenia uczestników jest zakończony, GA inicjalizuje negocjację, w której biorą udział kupcy (BA) i sprzedawca (Seller Agent; SeA). Po zakończeniu negocjacji SeA przekazuje (przez GA) informacje do SA o zwycięzcy negocjacji (jeżeli taki istnieje). Na polecenie SA kolejny agent sklepowy – Warehouse Agent (WA) – rezerwuje towar na określony czas, aby kupiec mógł się zastanowić, czy kupuje towar w tym sklepie. Jeżeli decyzja jest pozytywna, SA i CA uczestniczą w trzeciej fazie procesu kupna/sprzedaży – finalizacji zakupu.

Zanim przedstawimy dokładny opis systemu, zauważmy, że choć istnieje wiele prac na temat agentów w e-commerce oraz negocjacji między agentami, to jednak nasz projekt różni się od nich pod kilkoma istotnymi względami. 1) Zwykle rozważa się prowadzenie tylko jednej negocjacji. Nas interesuje scenariusz bardziej zbliżony do rzeczywistości, kiedy wiele produktów tego samego typu sprzedaje się jeden po drugim, czyli mamy do czynienia z sekwencją negocjacji. 2) Proces negocjacji zaprojektowaliśmy odmiennie do tego przedstawianego w literaturze. Zwy-

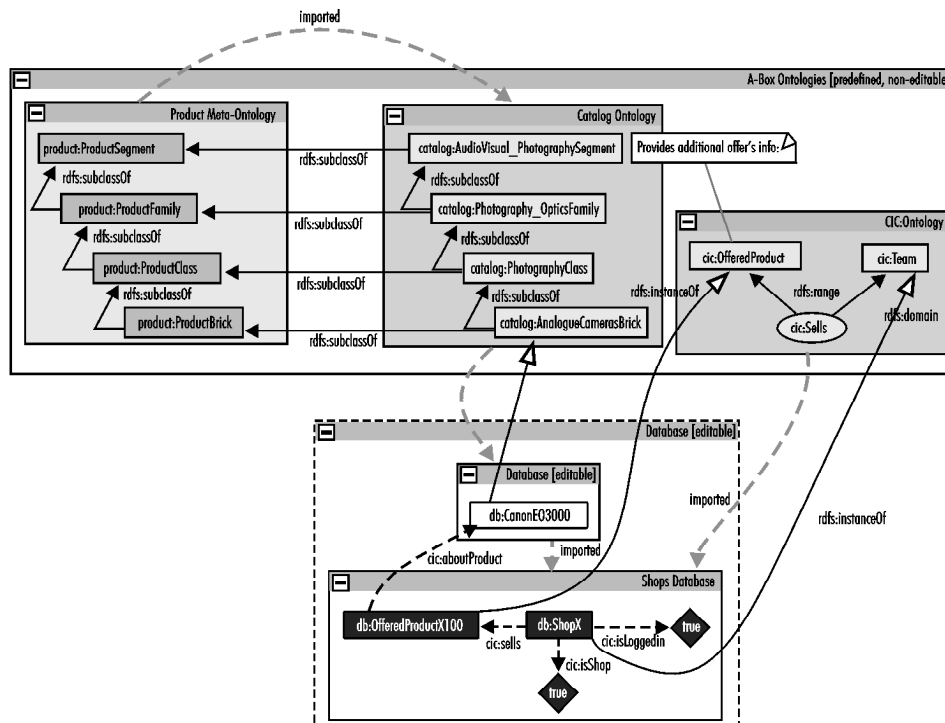
kle sprzedaż produktu ma miejsce w trakcie aukcji, do której nowi agenci przystępują w różnych momentach czasowych. W naszym systemie negocjacje są procesem dyskretnym. Mają one miejsce, gdy zgromadzi się pewna liczba kupców pragnących nabyć dany produkt. W międzyczasie zbiera się nowa grupa agentów BA zainteresowanych tymże towarem. 3) Skoro mamy do czynienia z sekwencją negocjacji cenowych, możliwa jest zmiana ich mechanizmu. Na przykład, pierwsze 127 sztuk zielonych szelek zostało sprzedanych w aukcji holenderskiej, a pozostałe 43 po ustalonej cenie z dużą zniżką z uwagi na koniec kolekcji. 4) Modelujemy system e-commerce całościowo, obejmując również to, co odbywa się przed i po negocjacjach (za wyjątkiem dostawy towaru i odbioru należności). 5) Mimo, że mobilność agentów często uznawana jest za ważny aspekt e-commerce, to rzadko wspomina się o problemie transportowania agenta z „dużą inteligencją”. W naszym systemie proponujemy rozwiązanie tego problemu wykorzystujące modułarną budowę agentów i precyzyjne określenie jakie moduły mają zostać przesłane, kiedy, przez kogo i dokąd.

Aby mogło dojść do zawarcia transakcji konieczne jest, aby uczestnicy procesu wiedzieli „o czym mówią”. Nasze rozważania zaczniemy więc od omówienia reprezentacji informacji.

11.3. Reprezentacja informacji i zarządzanie nią

Zdecydowaliśmy, że produkty sprzedawane w systemie, będą reprezentowane ontologicznie przy użyciu OWL Lite [32]. Do gromadzenia i przetwarzania danych ontologicznych wykorzystujemy środowisko Jena [21]. Z uwagi na to, że naszym celem nie jest stworzenie realistycznej ontologii produktów, ale pokazanie zasad operowania danymi w postaci ontologicznej, postanowiliśmy użyć stosunkowo prostych struktur. Na rysunku 11.2 przedstawiamy schemat ontologii oraz schemat zarządzania nią przez centrum informacji (CIC).

Kręgosłup ontologii produktów stanowi czterostopniowa hierarchia grup produktów: segment, rodzina, klasa i cegła (segment, family, class i brick). W ramach tej hierarchii poszczególne katalogi, jak na przykład katalog produktów audio-wizualnych, organizują grupy produktów danego typu. Każdy indywidualny produkt stanowi instancję odpowiedniej klasy cegły z wybranego katalogu. Pomysł oraz większość danych zostało zaadaptowanych ze standardu Global Product Classification [15].



Rysunek 11.2. Zarządzanie informacją przez CIC

Źródło: opracowanie własne.

Poniżej prezentujemy przykład instancji ontologii w formie N3. Reprezentuje ona ubranie jednorazowego użytku dla kobiet, chroniące kolana, wykonane z mieszanego materiału i odporne na zimno.

```

@prefix cc:
<http://www.ibspan.waw.pl/e-cap/product/clothing-
catalog.owl/#> .
@prefix prod:
<http://www.ibspan.waw.pl/e-cap/product/product.owl/#> .
@prefix :
<http://www.ibspan.waw.pl/e-cap/db/cic/products.owl/#> .

:Product1165510834375 a cc:ProtectiveLowerBodyWearBrick,
a prod:ProductEntity ;
cc:hasConsumerLifestage cc:Unclassified ;
cc:hasGender cc:Female ;
cc:hasIfDisposable cc:Yes ;
cc:hasSafetyFeatures cc:ColdResistant ;
cc:hasTypeOfMaterial cc:Combination ;
cc:hasTypeOfProtectiveLowerBodyWear
cc:ProtectiveKneePads .
    
```

Jak wspomnieliśmy, agent SA rejestruje w centrum informacji (CIC) wszystkie sprzedawane w swoim sklepie produkty (wysyła wiadomość zawierającą zserializowane dane towarów). Z tego względu CIC zapisuje informacje o produktach w formie ontologii rozszerzonej o dane oferującego SA (dokładniej, agenta GA). Dokładnie rzecz biorąc, obok zespołu ontologii produktów istnieje ontologia definiująca związek zarejestrowanych sklepów z produktami dostępnymi w bazie, za pomocą relacji *sells*. Na przykład, na rysunku 11.2 *sklep X* sprzedaje *Offered-Product100*, będący aparatem fotograficznym Canon EOS 3000.

Warto przy tej okazji wskazać na kilka przyjętych rozwiązań. Po pierwsze w naszym podejściu używamy dwóch różnych języków ontologii: FIPA ACL/SL [13] dla opisu operacji, o których wykonanie CIC jest proszony oraz OWL Lite do opisu produktów. Po drugie, dla uproszczenia niektórych procesów zdecydowaliśmy się na reprezentowanie różnych wariantów jednego produktu, jako osobne byty. Na przykład jeżeli pewne buty występują w rozmiarach 39, 40 oraz 41, to w bazie CIC będą reprezentowane jako trzy osobne rekordy. Po trzecie przyjęliśmy założenie, że wszyscy agenci przetwarzający informacje o produktach, będą znali ontologię produktów.

Kiedy użytkownik poprosi CA o zakup towaru, wówczas takie zamówienie tłumaczone jest na zapytanie SPARQL [38], które jest wykonywane przez CIC w środowisku Jena.

```

BASE <http://www.ibspan.waw.pl/e-cap/>
PREFIX product: <product/product.owl#>
PREFIX team: <cic/auxiliary/team.owl#>
PREFIX audio-video: <product/audio-visual-catalog.owl#>
SELECT ?product ?shop
FROM <db/cic/products.owl#>
FROM NAMED <db/cic/teams.owl#>
WHERE {
  ?product
    audio-video:hasFormatOfPhotographicFilm
      audio-video:_35Mm ;
    audio-video:hasIsfWithBuilt_InFlash audio-video:No ;
    audio-video:hasIsfWithZoom audio-video:No ;
    audio-video:hasIsfWithAutoFocus audio-video:Yes ;
    audio-video:hasTypeOfCamera
      audio-video:SingleLensReflex_Slr_ ;
    rdf:type audio-video:AnalogueCamerasBrick ;
    rdf:type product:ProductEntity .
  GRAPH <db/cic/teams.owl#> {
    ?shop team:sells ?offered .
    ?offered team:aboutProduct ?product .
  }
}
LIMIT 100

```

11.4. Negocjacje – wprowadzenie

Wiedząc już, jak w systemie opisywana jest informacja o produktach i ich sprzedawcach, możemy przejść do kluczowego zagadnienia – negocjacji cenowych. W ogólności, negocjacje są formą znajdowania kompromisu, w ramach którego zaangażowane strony akceptują porozumienie, np. w sprawie ceny towaru, terminu dostawy czy zawarcia pokoju [22, 30]. Jedną z najbardziej znanych form negocjacji cenowych są aukcje. Przedmiotem aukcji mogą być towary (dzieła sztuki, nieruchomości, etc.) lub usługi (naprawa budynku, budowa drogi, etc.). W praktyce spotyka się szeroką gamę aukcji, z których większość można dostosować do e-commerce. Dla celów naszego projektu ograniczyliśmy się do aukcji, które spotykamy w negocjacjach *przedsiębiorstwo-do-konsumentów* (B2C), pomijając transakcje typu *przedsiębiorstwo-do-przedsiębiorstw(a)* (B2B), w których możliwe są na przykład aukcje odwrotne (aukcje, w których klient ogłasza chęć zakupu towaru, a dostawcy zgłaszają swoje propozycje). Najbardziej popularnymi formami negocjacji cenowych są:

Aukcja angielska (English auction) – dotyczy sprzedaży jednego produktu (lub grupy produktów traktowanych jako jeden produkt). Bierze w niej udział sprzedawca i minimum dwóch kupujących. Uczestnicy składają propozycje zakupu „przebijając” poprzednią najwyższą cenę o co najmniej pewną określoną wartość (kwotę przebiccia). Aukcja kończy się, gdy nie ma już chętnych do dalszego podniesienia ceny (przez określony czas nie została zgłoszona nowa propozycja) lub gdy upłynie ustalony termin (aukcja kończy się o 11.54). To pierwsze rozwiązanie jest bardziej popularne w domach aukcyjnych, natomiast to drugie w przypadku aukcji internetowych. Sprzedający wyznacza cenę wywoławczą i opcjonalnie kwotę minimalną, poniżej której nie zgadza się na sprzedaż. Zwycięzcą aukcji jest kupujący, który złożył najwyższą ofertę (jeżeli nie była ona niższa od ceny minimalnej). Jeśli kwota minimalna nie jest ustalona, przyjmuje się, że jest to cena wywoławcza.

Aukcja holenderska (Dutch auction) – w chwili obecnej istnieje wiele schematów aukcji holenderskiej [7, 16]. Aukcje tego rodzaju rozpoczyna sprzedawca wysoką ceną wywoławczą, która jest następnie obniżana do momentu zaakceptowania ogłoszonej ceny przez któregokolwiek z klientów. Aukcja holenderska może być wykorzystana do sprzedaży jednego, jak i wielu produktów. W tym drugim przypadku kupujący określa ile przedmiotów w danej cenie chce kupić. Pozostałe przedmioty uczestniczą w kontynuacji aukcji, która przebiega w ten sam sposób. Warto zauważyć, że w *aukcji holenderskiej* sprzedawca jest stroną aktywną, natomiast kupujący są pasywni (poza akceptacją wybranej ceny). Jest to przeciwieństwo aukcji angielskiej, gdzie to kupujący byli stroną aktywną, a sprzedawca pasywną.

Przetargi (sealed bid first price auction) – uczestnicy składają oferty w „zalkowanych kopertach”. Zwycięzcą jest ten, kto zaoferował najwięcej (jeśli jest to kwota wyższa od kwoty minimalnej sprzedawcy).

Przetargi drugiej ceny (sealed bid second-price auction) – czasem zwana także aukcją *Vickrey'a* (nazwaną tak od nazwiska autora pomysłu Williama Vickrey'a). Jest to wariant *przetargu*, w którym zwycięzca płaci drugą co do wysokości zaproponowaną cenę, a nie najwyższą.

Negocjacje z ceną ustaloną (fixed-price negotiation) – sprzedający określił stałą cenę produktu. Kupujący mogą nabyć dowolną (dostępną) ilość towaru po tejże cenie.

11.5. Autonomiczne negocjacje oraz modułarna budowa agentów

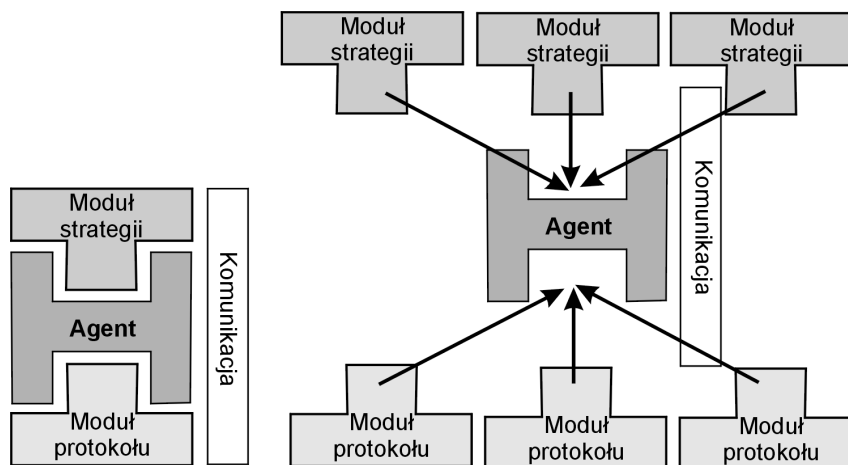
Zobaczmy teraz, jak można negocjacje realizować w systemach agentowych. Po wszechnie przyjmuje się, że mechanizm negocjacji wymaga dwóch elementów [22, 30]:

- *protokół negocjacyjny* – jest to konwencja, według której przebiega proces negocjacji,
- *strategia negocjacji* – zachowania podejmowane podczas negocjacji, mające doprowadzić do osiągnięcia pożądaných wyników.

Ponieważ negocjacje mają być prowadzone autonomicznie przez agentów programowych, konieczna jest formalizacja (w celu późniejszej reprezentacji algorytmicznej). Podsumowanie różnych podejść do formalizacji negocjacji można znaleźć w pracy [5, 26, 29, 42].

W proponowanym systemie wykorzystujemy rezultaty przedstawione przez Bartoliniego i jego zespół w [8, 9]. Zaproponowali oni strukturę dla implementacji negocjacji agentowych. Składa się ona z: 1) infrastruktury negocjacji, 2) ogólnego protokołu negocjacji oraz 3) reguł negocjacji. *Infrastruktura negocjacji* opisuje role uczestników oraz gospodarza (*host*) – zarządcy negocjacji. *Ogólny protokół negocjacji* definiuje trzy fazy negocjacji: przyjęcie do negocjacji, wymianę ofert oraz formułowanie porozumienia. Określa on, na przykład, kiedy i jakiego rodzaju wiadomości muszą wymienić gospodarz i uczestnicy negocjacji. *Reguły negocjacji* wymuszają mechanizm negocjacji i zostały podzielone na: reguły przyjmowania uczestników do negocjacji, reguły sprawdzania poprawności ofert, reguły do przestrzegania protokołu, reguły zmiany stanu negocjacji i informowania uczestników, reguły formułowania porozumienia oraz reguły zakończenia negocjacji.

W swoich pracach Bartolini i zespół zaproponowali, że w negocjacjach uczestniczą agenci mobilni, którzy przenoszą ze sobą całą wiedzę potrzebną do negocjacji. Ponieważ w naszym systemie negocjacje mogą się odbywać według wielu różnych scenariuszy, musieliśmy zmodyfikować takie podejście. A mianowicie, proponujemy rozwiązanie oparte na pracy [1], gdzie mobilny agent zbudowany jest ze szkieletu oraz modułów ładowanych dynamicznie na żądanie (więcej w [34, 35]).



Rysunek 11.3. Modułarna budowa agenta

Źródło: [35]

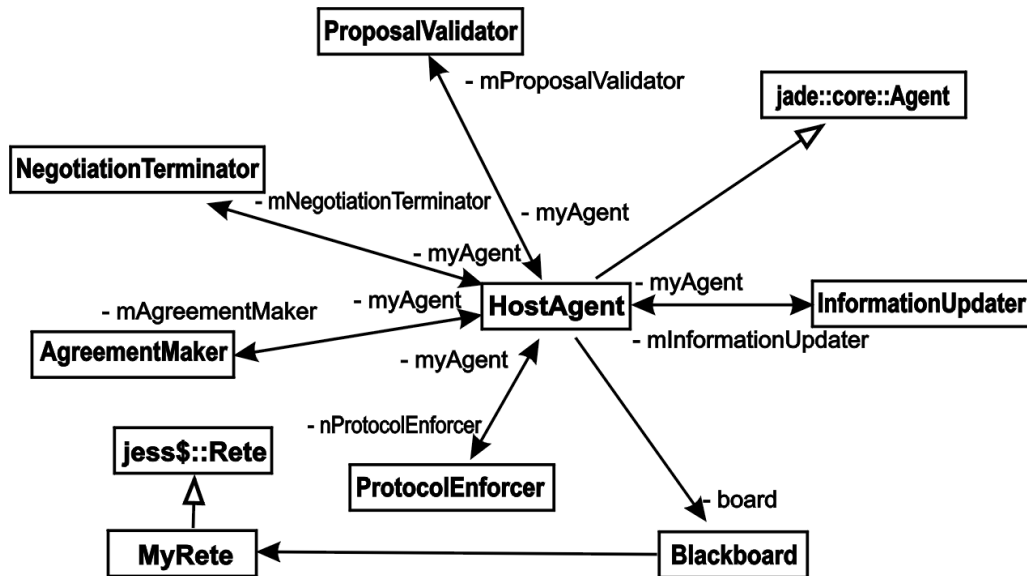
Pomysł zilustrowany jest na rysunku 11.3 przedstawiającym następujące elementy modularnego agenta:

- *Moduł komunikacji* – odpowiedzialny za komunikację między agentami; jest to moduł statyczny posługujący się językiem FIPA ACL [17] (jednak zawartość wiadomości może ulegać zmianie w zależności od miejsca negocjacji; na przykład negocjacje w e-sklepie w Chinach mogą wymagać specjalnego modułu komunikacji).
- *Moduł protokołu* – zawiera ogólne zasady negocjacji oraz specyficzne dane dotyczące negocjacji, która będzie miała miejsce (szablon negocjacji; negotiation template); informacje te są dostarczane przez agenta GA przed przystąpieniem do negocjacji;
- *Moduł strategii* – zawiera specyficzną dla danej negocjacji taktykę agenta, otrzymaną od agenta CA.

W ten sposób przez sieć przemieszcza się tylko minimalny szkielet agenta (ew. z modułem komunikacyjnym) natomiast pozostałe moduły zostają przyłączone już w sklepie.

11.6. Negocjacje oparte na regułach

Załóżmy teraz, że agent BA został skompletowany i jest gotowy do uczestnictwa w negocjacjach cenowych. Jak wspomniano *reguły negocjacji* wykorzystywane są do wymuszania konkretnego rodzaju negocjacji cenowych i dzielą się na: a) reguły przyjmowania uczestników do negocjacji, b) reguły poprawności ofert, c) reguły ustalania zgodności z protokołem, d) reguły uaktualniania stanu negocjacji oraz informowania uczestników, e) reguły zawierania porozumienia oraz f) reguły zakończenia negocjacji. Na podstawie takiego podziału, w pracach [8, 9], zaproponowano utworzenie po stronie gospodarza następujących komponentów: Gatekeeper, Proposal Validator, Protocol Enforcer, Information Updater, Negotiation Terminator oraz Agreement Maker. W związku z wykorzystywanym przez nas mechanizmem obsługi negocjacji, *Gatekeeper* stał się pełnoprawnym agentem z zadaniami opisanymi powyżej. Tak więc gospodarz negocjacji ma teraz strukturę przedstawioną na rysunku 11.4.



Rysunek 11.4. Struktura klas gospodarza negocjacji

Źródło: opracowanie własne.

Gospodarz negocjacji został zaimplementowany jako agent JADE i dziedziczy po klasie `jade.core.Agent`. Komponenty kontroli negocjacji zostały do niego włączone jako zwykłe klasy Javy, z których każda posiada metodę `handle()`, aktywowaną gdy należy sprawdzić reguły, za które odpowiada. Dodatkowo, co widać na

rysunku 11.4, gospodarz zawiera obiekt tablica informacyjna (*blackboard*). Obiekt ten jest silnikiem reguł JESS [23] (klasy *jess.Rete*) zainicjowanym regułami konkretnej negocjacji i uruchamia go każde sprawdzanie tych reguł.

11.6.1. Sprawdzanie reguł

Wszystkie komponenty gospodarza używają jednej współdzielonej instancji systemu eksperckiego JESS. Zaletą takiego rozwiązania jest fakt, że każdy host posiada wyłącznie jeden silnik reguł zamiast sześciu sugerowanych przez Bartoliniego (jeden dla każdego komponentu). Tak więc w przypadku m gospodarzy mamy m instancji JESS zamiast $6m$; zyskując tym samym na wydajności oraz ograniczając ilość wymaganej pamięci.

Reguły i fakty zarządzane przez silnik reguł podzielone są na moduły JESS. Obecnie jeden moduł odpowiada za dane na tablicy informacyjnej, a osobne za reguły używane przez poszczególne komponenty. Fakty dla tablicy informacyjnej są wyrażeniami JESS *deftemplate* i mogą reprezentować, między innymi (pełna lista faktów zależy od rodzaju negocjacji): aktualną ofertę sprzedaży, niewidoczną dla kupców minimalną kwotę sprzedawcy, maksymalny czas oczekiwania na następną ofertę, kwotę aktualnie najwyższej oferty, etc.

11.6.2. Realizacja negocjacji – aukcja angielska

Aukcja angielska jest jedną z najbardziej popularnych obecnie metod negocjacji cenowych. Głównym powodem takiego stanu rzeczy jest jej prostota i fakt, iż łatwo daje się realizować elektronicznie (używają jej, m.in. eBay.com i Allegro.pl).

W naszym systemie, w przypadku aukcji angielskiej *szablon negocjacji* będzie zawierał następujące informacje:

- numer identyfikacyjny negocjacji,
- numer identyfikacyjny produktu,
- minimalną kwotę przebiccia,
- maksymalny czas braku aktywności w trakcie aukcji, po którym następuje zakończenie aukcji,
- cenę początkową (szablon sprzedawcy),
- minimalną zarezerwowaną cenę sprzedawcy (szablon sprzedawcy).

Strategia klienta w przypadku aukcji angielskiej mówi kiedy ogłosi on swoją ofertę oraz jaka będzie jej wysokość. Oczywiście wartość oferty musi być wyższa

od aktualnej ceny maksymalnej co najmniej o kwotę przebicia podaną w *szablonie*. Strategia zawiera także informację, ile maksymalnie klient jest gotów zapłacić za dany towar (górną granicę dla ofert zgłaszanych przez BA). Strategii dla sprzedającego w aukcji angielskiej nie ma, ponieważ w tym przypadku jest on stroną pasywną.

Poniżej przedstawiamy dwie przykładowe reguły występujące w aukcji angielskiej.

- OFERTA–KUPIEC – reguła ta mówi, że kupiec może wysłać ofertę wówczas, gdy istnieje oferta sprzedającego (negocjacja została zapoczątkowana)

<p>JEŻELI Jest poprawna oferta Pr uczestnika w roli kupca ^ Istnieje aktywna oferta uczestnika w roli sprzedawcy</p> <p>WÓWCZAS Oferta Pr jest ogłaszana</p>
--

- PRZEBICIE–KUPIEC – zgłaszana przez kupca oferta musi być wyższa od aktualnej oferty co najmniej o kwotę przebicia

<p>JEŻELI Kwota przebicia to Inc ^ Aktualna najwyższa oferta to B ^ Ofertę Pr na towar A z ceną P złożył kupiec ^ $P \geq B + Inc$</p> <p>WÓWCZAS Oferta Pr jest aktywna</p>

11.6.3. Realizacja negocjacji – aukcja holenderska

Zajmijmy się teraz nieco bardziej skomplikowanym mechanizmem negocjacji, jakim może być wieloproduktowa aukcja holenderska. Opis przyjętej przez nas wersji aukcji holenderskiej znajduje się w punkcie 11.4. Dodajmy tylko, że przyjęliśmy dodatkowe ograniczenie, że jeden kupujący może zgłosić co najwyżej jedną poprawną ofertę kupna.

Dla *aukcji holenderskiej* szablon negocjacji musi zawierać następujące ustalenia:

- numer identyfikacyjny negocjacji,
- numer identyfikacyjny produktu,

- minimalną kwotę przebicia, o którą sprzedawca musi zmniejszać cenę przy każdej kolejnej ofercie,
- maksymalny czas braku aktywności w trakcie aukcji, po którym następuje zakończenie aukcji,
- liczbę sztuk N towaru wystawioną na sprzedaż,
- cenę początkową (szablon sprzedawcy),
- minimalną zarezerwowaną cenę sprzedawcy (szablon sprzedawcy).

W przypadku aukcji holenderskiej, strategia kupującego określa, kiedy zaakceptować cenę sprzedawcy. Najprostszą strategią jest sprawdzenie, czy aktualna cena jest niższa od maksymalnej, którą klient jest skłonny zapłacić i jeśli tak jest, to zaakceptować ofertę. Można jednak oprzeć strategię na upływie czasu, ilości pozostałych w ofercie przedmiotów, czy na tempie, w jakim ich ubywa. Strategia sprzedawcy natomiast powinna zawierać następujące parametry:

- cenę początkową,
- funkcję wyznaczania przedziału czasu, po którym sprzedawca ogłosi kolejną ofertę (czas ten musi być krótszy niż maksymalny czas braku aktywności, który kończy negocjację),
- cenę minimalną, poniżej której sprzedawca nie będzie licytował.

A oto przykładowa reguła dla aukcji holenderskiej. Kupiec może wysłać ofertę jedynie wówczas, kiedy istnieje aktywna oferta sprzedawcy. Ponadto oferowana przez kupca cena (funkcja X) musi być równa aktualnej cenie sprzedaży oraz żądana ilość (funkcja M) nie może być większa od ogólnie dostępnej puli towaru. Powyższe warunki sprawdza następująca reguła:

➤ AKCEPTOWANIE–KUPIEC

JEŻELI

oferta Pr pochodzi od uczestnika w roli kupca \wedge

aktywna oferta sprzedawcy jest na kwotę B \wedge

$X(Pr) = B$ \wedge

$M(Pr) \leq N$

WÓWCZAS

oferta Pr jest akceptowana.

11.7. Zakończenie negocjacji

Każdorazowe zakończenie negocjacji przez jednego z agentów BA prowadzi do przesłania informacji o jej rezultatach do agenta CA. Wiadomości te są obsługiwane w kolejności wynikającej z czasu rezerwacji (oraz kwestii związanych z zaufaniem; zobacz punkt 11.9). Na przykład, jeśli dana rezerwacja kończy się w ciągu następnych 37 sekund to trzeba podjąć decyzję czy wykorzystać ją i dokonać zakupu [14]. W trakcie trwania analizy mogą zostać nadesłane wyniki negocjacji, które miały miejsce w innych sklepach i mogą wpłynąć na jej przebieg. Ogólnie rzecz biorąc, rezultatem analizy może być: 1) decyzja o realizacji zakupu w jednym ze sklepów, 2) decyzja o oczekiwaniu na lepszą propozycję (w tym rezygnacja z pewnej grupy rezerwacji i nakazanie agentom przystąpienia do ponownych negocjacji), lub 3) poinformowanie klienta, że transakcja jest niemożliwa do przeprowadzenia (na przykład dlatego, iż z informacji otrzymanych od BA wynika, że ceny rynkowe - ceny stałe i wyniki negocjacji - są wyższe od zadanych warunków cenowych).

Zauważmy, że podjęcie decyzji o wyborze oferty nie musi zakończyć się sukcesem. Komunikacja z agentem znajdującym się na innym komputerze może trwać tak długo, że rezerwacja zostanie anulowana. Fakt ten musi być brany pod uwagę przez CA, gdy analizuje on posiadane w danym momencie informacje dotyczące obsługi konkretnego zlecenia.

Kiedy próba zakupu powiedzie się, jak również jeżeli podjęta zostaje decyzja o „niewykonalności” danego zlecenia, CA wysyła wiadomości do wszystkich BA pracujących nad danym zleceniem, aby zakończyli swoje działanie.

Wszelkie dane o zdarzeniach mających miejsce podczas obsługi poleceń klienta są zbierane i przechowywane na przyszłość. Mowa tutaj, na przykład, o: odmowie przyjęcia do negocjacji, wynikach negocjacji (pozytywnych i negatywnych), uzyskanej cenie i długości rezerwacji. Efektem analiz z wykorzystaniem takich informacji może być, na przykład: (1) wykluczenie z kręgu zainteresowań sklepu, który systematycznie oferuje towary po „zawyżonych” cenach (na podstawie wyników aukcji/proponowanych stałych cen), (2) ocena wartości rynkowej towaru (przebrane i wygrane aukcje dostarczają takiej informacji) (3) ocena stopnia zaufania ze strony SA (na podstawie zmiany długości rezerwacji w danym sklepie), (4) ocena prędkości sprzedaży (krótsza rezerwacja „bez powodu” może oznaczać, że towar się dobrze sprzedaje), (5) zmiana strategii negocjacji (sekwencja przegranych negocjacji danego typu skłania do zmiany strategii).

Z drugiej strony po zakończeniu negocjacji SA dokonuje rezerwacji towaru u WA, informuje zwycięskiego BA o przyznanej długości rezerwacji i oczekuje na dalszy rozwój sytuacji:

1. Jeśli zwycięski BA potwierdza rezerwację, SA sprawdza u WA status rezerwacji. Jeżeli rezerwacja nie wygasła, wówczas SA potwierdza transakcję. Ta wia-

domość rozpoczyna finalizację transakcji obejmującą, m.in. płatności i dostawę. Jeżeli rezerwacja wygasła, SA wysyła do BA informację o odrzuceniu.

2. Jeśli CA odwołuje rezerwację (za pośrednictwem BA), wówczas SA poleca WA wycofać odpowiednią rezerwację i modyfikuje zaufanie do danego CA.

Przedstawione scenariusze zamykają określony proces w SA, jednak przez cały czas ma on pieczę nad pozostałymi transakcjami i negocjacjami. Ponadto cyklicznie przeprowadza on analizę sytuacji dotyczącej danego towaru, co może owocować zmianami w sposobie (mechanizm negocjacji) lub parametrach (ceny minimalnej, itp.) jego sprzedaży. Na przykład, gdy zostały 22 sztuki produktu, wówczas może zapaść decyzja o wyprzedzący po cenie stałej. W każdym z takich przypadków powstaje odpowiedni szablon, który trafia do GA.

Podobnie do CA, SA też zapisuje w swojej bazie wiedzy wszelkie informacje dotyczące wszystkich wydarzeń w sklepie, np.: wyniki negocjacji (pozytywne i negatywne), informacje o zwycięzcach, kto potwierdził rezerwację, a kto zrezygnował z zakupu, itd. Informacje zgromadzone w tejże bazie mogą spowodować, np.: (1) niedopuszczenie niektórych BA/CA do negocjacji, (2) zmianę mechanizmu negocjacji cenowych, (3) zmianę parametrów negocjacji (np. kwoty minimalnej sprzedawcy), zmianę czasu rezerwacji (towary, które się bardzo dobrze sprzedają, mają krótszy czas rezerwacji).

11.8. Zarządzanie czasem

Zarządzanie czasem odgrywa istotną rolę w proponowanym systemie. Po pierwsze wiele typów negocjacji cenowych wykorzystuje czas, np. do kiedy można ogłosić nową ofertę lub do kiedy sprzedawca czeka na oferty. Ponieważ jednak negocjacje cenowe odbywają się na jednym komputerze (nie ma negocjacji na odległość), wszyscy uczestnicy mogą korzystać z zegara tej samej maszyny.

Drugi przypadek, to rezerwacja produktu i tutaj sprawa się komplikuje, ponieważ dotyczy wielu komputerów. Na przykład, jeżeli CA wysłał 10 agentów BA na 10 różnych maszyn, wówczas musimy sobie radzić z 11 różnymi czasami. Na przykład, gdy jeden komputer spieszy się o 13 sekund w stosunku do GMT, inny może opóźnić się o 17 sekund. Załóżmy, że część z tych agentów wygrała negocjacje i przekazała do CA informacje kiedy wygasają rezerwacje. W tej sytuacji CA musi wiedzieć dokładnie ile ma czasu na podjęcie decyzji, czy potwierdzić chęć zakupu, zanim poszczególne rezerwacje wygasną.

Proponowane rozwiązanie wykorzystuje czas uniwersalny. W tym celu użyliśmy protokołu czasu sieciowego SNTP [18, 19, 20], aby ustalić czas uniwersalny i różnicę czasową między tym czasem a lokalnym zegarem na komputerze. Wykorzystując ten protokół, każdy CA i GA ustala taką różnicę podczas inicjalizacji.

Zakładamy przy tym, że lokalny zegar działa poprawnie, i że proces ustalania różnicy czasu nie będzie musiał być uruchamiany zbyt często (możliwe jest również odświeżanie różnicy czasowej przez GA przed każdą negocjacją). Tak ustalona różnica czasowa jest przekazywana do BA razem z szablonem negocjacji.

Po wygranej licytacji, BA przesyła różnicę czasową do CA razem z godziną wygaśnięcia rezerwacji (np. „rezerwacja jest ważna do 12:35:55, a lokalna różnica czasowa to –00:00:01”). Wówczas CA może dowiedzieć się dokładnie ile ma czasu na analizę (np. jeżeli sklep późni się o 10 sekund względem GMT, a klient spieszy się o 5 sekund względem GMT, wówczas CA względem GA spieszy się o 15 sekund). Oczywiście opóźnienia sieciowe mogą odegrać kluczową rolę podczas potwierdzania rezerwacji. W przypadku wolniejszego połączenia sieciowego, wiadomość o decyzji klienta powinna zostać wysłana odpowiednio wcześniej, a ustalanie właściwego momentu leży w gestii agenta CA.

11.9. Zarządzanie zaufaniem

Ostatnim tematem, który chcielibyśmy omówić, jest zarządzanie zaufaniem. Zauważmy tutaj, że rozpatrywać można dwa bliskie sobie pojęcia: zaufanie i reputację. W literaturze pojęcia te rozumiane są najczęściej jako: **zaufanie** – wiara jednostki w możliwości, uczciwość oraz niezawodność (drugiej jednostki), *na bazie własnych dotychczasowych doświadczeń*; **reputacja** – wiara jednostki w możliwości, uczciwość oraz niezawodność (drugiej jednostki), *na podstawie rekomendacji innych jednostek*. Tak więc zaufanie jest zwykle traktowane jako relacja *jeden-do-jednego*, natomiast reputacja jako *jeden-do-wielu*. Równocześnie oba związki budowane są na podstawie długotrwałych doświadczeń.

Istnieje wiele sposobów obliczania reputacji i zaufania. Podamy tutaj tylko listę przykładów, a po konkretne rozwiązania odsyłamy zainteresowanych czytelników do wskazanej literatury: 1) suma lub średnia rankingów [36], 2) systemy Bayesa [39], 3) model dyskretny [2, 3, 10, 11], 4) modele rozmyte [31] i inne [24, 25, 33, 43, 44].

Choć możliwe byłoby wprowadzenie do naszego systemu pojęcia reputacji, w chwili obecnej wymiana informacji między klientami lub sprzedawcami nie jest przewidziana. Z tego powodu zajmujemy się wyłącznie zaufaniem.

11.9.1. Zaufanie po stronie klienta

Działanie agenta CA, reprezentującego użytkownika, opiera się na historii kontaktów ze sklepami. Obecnie zakładamy, że negocjacje są fair (np. bez sztucznego podbijania ceny przez podstawionych agentów lub umów między sklepami oferującymi taki sam towar) oraz, że agenci nie mogą ukryć swojej tożsamości: mogą otrzymać

nowe ID, aby wymazać brak zaufania do siebie w niektórych sklepach, jednak nie mogą podszywać się pod kogoś innego, kto cieszy się wysokim zaufaniem.

Większość sytuacji, mogących wpływać na zaufanie do sklepu, ma miejsce podczas finalizacji transakcji (inne można znaleźć w [6]). Proces ten zawiera, m.in. płatność oraz dostawę. Wszelkie opóźnienia, zła jakość dostarczonego towaru, przysłanie niewłaściwego towaru, itd. wpływają negatywnie na zaufanie do takiego dostawcy.

Zastanówmy się teraz, jak wpływa to na zachowanie klienta. Kiedy użytkownik prosi CA o zakup pewnego produktu, CA otrzymuje od CIC listę sklepów, które ten produkt oferują. Taka lista jest następnie przetwarzana przez CA, który między innymi ustala stopień zaufania do każdego e-sklepu (proces każdorazowego dostosowywania listy wynika z zastosowania modelu zaufania z zapominaniem). Jeżeli okaże się, że na liście znajdują się sklepy z zaufaniem poniżej pewnej dopuszczalnej wartości, wówczas są one wykluczane z kręgu zainteresowań. Oczywiście wyjątkiem od tej reguły może być sytuacja, kiedy interesujący produkt oferuje niewielu sprzedawców. W takim przypadku klient może zechcieć zaryzykować zakupy w sklepie, o którym ma nienajlepsze zdanie.

Jeżeli jednak po wykluczeniu niechcianych sklepów lista ciągle jest długa, wówczas CA może zastosować wielokryterialny ranking (np. używając metody Saaty'go [37]) i na tej podstawie wybrać oferty najbardziej go interesujące. Zauważmy, że wtedy może mieć znaczenie, że dany sklep został uznany przez klienta za drogi (tzw. wizerunek), ponieważ każdemu kryterium należy przyporządkować pewną wagę. Jeden klient może powiedzieć: „wolę drogie produkty wysokiej jakości od tanich”, kiedy inny powie: „szukam przede wszystkim tańszych produktów z nie koniecznie ekspresową dostawą”.

Można by zapytać, dlaczego nie wysyłać agentów BA do wszystkich sklepów. Odpowiedź jest również powiązana z zaufaniem. Za każdym razem, kiedy BA wygrywa aukcję i jej nie finalizuje, sklep zmniejsza do niego zaufanie. Zatem w uproszczeniu, jeżeli klient wysłałby N kupców, z których K wygrałoby aukcje, to przy finalizacji jednej transakcji, w jednym sklepie nasze zaufanie by wzrosło, a w $K-1$ spadło. Z uwagi na to, że niskie zaufanie może doprowadzić do niedopuszczenia do negocjacji, przy wyborze sklepów do odwiedzenia należy postępować bardzo ostrożnie.

11.9.2. Zaufanie po stronie sklepu

Jak opisano wcześniej, pierwszy kontakt między sklepem a klientem odbywa się za pośrednictwem agenta GA, który danego kupca wpuszcza lub nie. Decyzja ta opiera się na zaufaniu, które sklep przypisuje konkretnemu klientowi. Kiedy wartość zaufania zejdzie poniżej pewnego progu, wówczas GA nie pozwoli kupcowi na negocjacje.

Z punktu widzenia sklepu, realizacja rezerwacji odgrywa kluczową rolę w budowaniu zaufania do klienta (inne przypadki opisane są w [6]). Przypomnijmy, że

jeżeli BA wygrał negocjacje, nie znaczy to jeszcze, że kupi dany towar. Sklep jednak musi na pewien czas zarezerwować odpowiednią ilość produktu, wycofując ją tym samym z ogólnej puli dostępnej dla innych klientów. Skoro głównym zadaniem agenta SA jest sprzedaż jak największej ilości towaru, to klient, który sfinalizował transakcję, umacnia swoją pozycję w sklepie.

W przypadku, gdy BA nie dokona zakupu, mamy do rozpatrzenia dwa scenariusze. 1) BA stosunkowo wcześniej, w czasie trwania rezerwacji, poinformował o swojej rezygnacji (np. w pierwszych 2 minutach przy 59 minutach rezerwacji). SA oczywiście nie jest wówczas zadowolony z takiego obrotu sprawy, jednak takie są założenia naszego systemu i wszyscy użytkownicy muszą godzić się z konsekwencjami. Dlatego też w tej sytuacji kara nałożona na BA nie powinna być zbyt surowa (lub nie powinno jej być w ogóle). 2) Jeżeli jednak rezerwacja wygasa, wówczas sklep ma pełne prawo do utraty zaufania do takiego nabywcy. Nie dość bowiem, że na dłuższy czas pomniejszono dostępną dla sprzedaży ilość towaru, to jeszcze należy sobie uświadomić, iż drugi w kolejności licytujący mógłby kupić ten towar, lecz nie wygrał negocjacji. Obniżenie wiarygodności nierzetelnego kupca może wiązać się z zejściem jego zaufania poniżej wartości progowej, a to uniemożliwi mu ponowne wejście do negocjacji [4].

Należy również przewidzieć problemy podczas finalizacji transakcji. Powiedzmy BA wygrał negocjacje i potwierdził chęć zakupu, ale z różnych przyczyn CA nie przelał pieniędzy. W takim przypadku poziom zaufania do klienta również się obniża. Jednak w naszym systemie ten etap zakupu nie jest implementowany. Dlatego też pomijamy takie przypadki w szczegółowych rozważaniach.

11.9.3. Karanie i nagradzanie

Zaufanie klienta do sklepu wykorzystuje się wyłącznie w momencie wyboru sklepów, w których podjęta zostanie próba zakupu towarów. Natomiast zaufanie sklepu do klienta używane jest w dwóch przypadkach. Po pierwsze dany kupiec może być dopuszczony do negocjacji (po raz pierwszy, lub usiłując do niej powrócić). Jednakże znacznie ciekawszy jest wpływ zaufania na długość rezerwacji. Im większe bowiem zaufanie sklepu do kupca, tym dłużej sklep może poczekać na jego decyzję. Więcej na ten temat, jak również o przykładowym sposobie obliczania zaufania, można przeczytać w [6].

Uwagi końcowe

Celem powyższego rozdziału było podsumowanie naszych prac prowadzących do stworzenia modelowego agentowego systemu e-commerce. W chwili obecnej opisany system jest implementowany, a kody źródłowe znajdują się w repozytorium Sourceforge pod adresem: <http://e-cap.sourceforge.net>

Bibliografia

- [1] *A Plug-in Architecture Providing Dynamic Negotiation Capabilities for Mobile Agents* (1998) M. T. Tu, F. Griffel, M. Merz, W. Lamersdorf, Lecture Notes in Computer Science.
- [2] Abdul-Rahman A., Hailes S., *Supporting Trust in Virtual Communities*. In Proceedings of the Hawaii International Conference on System Sciences, Maui, Hawaii, 4–7 January 2000.
- [3] Audun J., Roslan I., Boyd C., *A Survey of Trust and Reputation Systems for Online Service Provision*. Decision Support Systems.
- [4] Bădică C., Bădită A., Ganzha M., Paprzycki M., *Developing a Model Agent-based E-commerce System*. In: Jie Lu et. al. (eds.) *E-Service Intelligence – Methodologies, Technologies and Applications*, Springer, 2006.
- [5] Bădică C., Bădită A., Ganzha M., Paprzycki M., *Implementing Rule-Based Automated Price Negotiation in an Agent System*. Journal of Universal Computer Science, in press. (2007)
- [6] Bădică C., Ganzha M., Gawinecki M., Kobzdej P., Paprzycki M., *Towards Trust Management in an Agent-based E-commerce System – Initial Considerations*. In: A. Zgrzywa (ed.) Proceedings of the MISSI 2006 Conference, Wrocław University of Technology Press, Wrocław, Poland (2006) 225–236.
- [7] Bădică C., Ganzha M., Gawinecki M., Kobzdej P., Paprzycki M.: *Utilizing Dutch Auction in an Agent-based Model E-commerce System* In: Proceedings of the 14th International Enformatika Conference, World Enformatika Society, 2006, 7–12.
- [8] Bartolini, C., Preist, C., Jennings, N.R., *A Software Framework for Automated Negotiation*; Proc. Of SELMS. Lect. Notes in Comp. Sci. 3390, Springer, Berlin (2005) 213–235.
- [9] Bartolini, C., Preist, C., Jennings, N.R., *Architecting for Reuse: A Software Framework for Automated negotiation*; Proc. of AOSE: Int. Workshop on Agent-Oriented Software Engineering, Bologna, Italy, Lect. Notes in Comp. Sci. 2585, Springer, Berlin, (2002) 88–100.
- [10] Cahill V., Shand B., Gray E., et al. *Using Trust for Secure Collaboration in Uncertain Environments*. Pervasive Computing, 2(3):52–61, July–September 2003.
- [11] Carbone M., Nielsen M., Sassone V., *A Formal Model for Trust in Dynamic Networks*. In Proc. of Int. Conference on Software Engineering and Formal Methods, Brisbane, September 2003.
- [12] *Ecommerce definition and types of ecommerce*, <http://www.digsmith.com/ecommerce-definition.html>
- [13] *FIPA SL Content Language Specification*, 2002, <http://www.fipa.org/specs/fipa00008/SC00008I.html>
- [14] Gawinecki M., Ganzha M., Kobzdej P., Paprzycki M., Bădică C., Scafes M., Popa Gabriel-George (2006) *Managing Information and Time Flow in an Agent-based E-commerce System*. In: D. Petcu et. al. (eds.), Proceedings of the Fifth International Symposium on Parallel and Distributed Computing, IEEE CS Press, Los Alamitos, CA, 2006, 352–359.
- [15] *Global Product Classification*, <http://www.gsl.org/productssolutions/gdsn/gpc/>
- [16] <http://glossary.global-investor.com/terms/Dutch-auction.htm?ginPtrCode=00000&id=12389&PopupMode=true> 39–40.
- [17] <http://www.fipa.org>
- [18] <http://www.ietf.org/rfc/rfc1361.txt>
- [19] <http://www.ietf.org/rfc/rfc1769.txt>
- [20] <http://www.ietf.org/rfc/rfc2030.txt>
- [21] *Jena 2 – A Semantic Web Framework*, Hewlett Packard, <http://www.hpl.hp.com/semweb/jena2.htm>
- [22] Jennings N. R., Faration P., Lomuscio A. R., Parsons S., Sierra C. And Wooldridge M.: *Automated Negotiation: Prospects, Methods and Challenges*, In: Int Journal of Group Decision and Negotiation, 2001, Keynote Paper.
- [23] *JESS: Jave Expert System Shell*; herzberg.ca.sandia.gov/jess/
- [24] Jøsang A. A., *Logic for Uncertain Probabilities*. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 9(3): 279–311, June 2001.

- [25] Jøsang A., *Trust-Based Decision Making for Electronic Transactions*. In: L. Yngström and T. Svensson, editors, Proceedings of the 4th Nordic Workshop on Secure Computer Systems (NORDSEC'99). Stockholm University, Sweden, 1999.
- [26] Kittsteiner T., Ockenfels A., *On the Design of Simple Multi-unit Online Auctions*, In: N. Jennings, G. Kersten, A. Ockenfels and C. Weinhardt (Eds.), Negotiation and Market Engineering, Dagstuhl Seminar Proceedings.
- [27] Kowalczyk R., et al.: *Integrating Mobile and Intelligent Agents*. In: *Advanced E-commerce: A Survey. Agent Technologies, Infrastructures, Tools, and Applications for E-Services*, Proceedings NODE'2002 Agent-Related Workshops, Erfurt, Germany, LNAI 2592, Springer Verlag, (2002) 295–313.
- [28] Kowalczyk R., Franczyk B., Speck A., *Inter-Market, towards intelligent mobile agent E-Market places*. 9th Annual IEEE International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS 2002), Lund, Sweden (2002), 268–276.
- [29] Lochner, K. M., Wellman, M.P.: *Rule-Based Specification of Auction Mechanisms*. Proc. AAMAS'04, ACM Press, New York, USA, (2004).
- [30] Lomuscio, A. R., Wooldridge, M., Jennings, N.R.: *A classification schema for negotiation in electronic commerce*; F. Dignum, C. Sierra (Eds.): *Agent Mediated Electronic Commerce: The European AgentLink Perspective*, Lect. Notes in Comp. Sci. 1991, Springer, Berlin (2002) 19–33.
- [31] Manchala D. W., *Trust Metrics, Models and Protocols for Electronic Commerce Transactions*. In Proceedings of the 18th International Conference on Distributed Computing Systems, 1998.
- [32] OWL – *Web Ontology Language Overview*, <http://www.w3.org/TR/owl-features/>
- [33] Page L., Brin S., Motwani R., Winograd T., *The PageRank Citation Ranking: Bringing Order to the Web. Technical report*, Stanford Digital Library Technologies Project, 1998.
- [34] Parakh G., Paprzycki M., Nistor C. E., *Dynamically Loaded Reasoning Models in Negotiating Agents*, Proceedings of the 3rd European E-COMM-LINE 2002 Conference, Bucharest, Romania, 2002, 199–203.
- [35] Parakh G., Sandhya R., Paprzycki M., Ajith A., Johnson Th. (2003) *Agents Capable of Dynamic Negotiations*. In: M. Paprzycki (ed.), *Electronic Commerce; Research and Development*, ACTEN Press, Wejherowo, Poland, 2003, 113–120.
- [36] Resnick P., Zeckhauser R., *Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System*. In M. R. Baye, editor, *The Economics of the Internet and E-Commerce*, volume 11 of *Advances in Applied Microeconomics*. Elsevier Science, 2002.
- [37] Saaty T.L. *The Analytik hierarchy process*, RWS Publications, 1990.
- [38] SPARQL Query Language for RDF, <http://www.w3.org/TR/rdf-sparql-query/>
- [39] Withby A., Jøsang A., Indulska J.: *Filtering Out Unfair Ratings in Bayesian Reputation Systems*. In Proceedings of the 7th Int. Workshop on Trust in Agent Societies. ACM, 2004.
- [40] Wooldridge M.: *An Introduction to MultiAgent Systems*, John Wiley & Sons, (2002), Laudon K. C., Traver C. G.: *E-commerce. business. technology. society* (2nd ed.). Pearson Addison-Wesley, (2004).
- [41] *WTO special study on e-commerce* www.wto.org/english/tratop_e/ecom_e/special_study_e.pdf
- [42] Wurman P. R., Wellman M. P., Walsh W. E.: *A Parametrization of the Auction Design Space; Games and Economic Behavior*, 35, 1/2, (2001) 271–37.
- [43] Yu B., Singh M. P.: *An Evidential Model of Distributed Reputation Management*. In Proceedings of the First Int. Joint Conference on Autonomous Agents & Multiagent Systems. ACM, July 2002.
- [44] Yu B., Singh P. M.: *A social mechanism of reputation management in electronic communities*. In Proceedings of Fourth International Workshop on Cooperative Information Peers, 154–165, 2000.