

Spis treści

Wstęp	9
-------------	---

Część I. Systemy

Iwona Chomiak-Orsa: Mapowanie procesów podstawowym etapem realizacji przedsięwzięcia informatycznego.....	13
Wiesława Gryncewicz: Identyfikacja procesów informacyjnych realizowanych w urzędach skarbowych w Polsce.....	21
Dorota Jelonek: Portal korporacyjny w zarządzaniu zasobami informacyjnymi o otoczeniu przedsiębiorstwa	32
Maja Leszczyńska: Możliwości zastosowania technologii czasu rzeczywistego w międzyorganizacyjnym systemie informacyjnym logistyki	42
Andrzej Niesler: Integracja systemów informatycznych przedsiębiorstwa w architekturze z autonomicznym rejestrem usług sieciowych.....	56
Monika Sitarska: Portale korporacyjne jako element systemu zarządzania informacją i wiedzą w organizacji	66

Część II. Metody

Damian Dziembek: Strategiczne implikacje dla organizacji gospodarczych wynikające z zastosowania wirtualnego outsourcingu informatycznego.....	79
Wiesława Gryncewicz: Analiza i ocena jakości zasobów informacyjnych w urzędach skarbowych w Polsce	96
Łukasz Łysik: Miary zastosowania technologii mobilnych w procesach handlowych.....	110
Adam Nowicki, Mariusz Nosal: Zasady ładu informatycznego w przedsiębiorstwie	121
Jolanta Pondel, Maciej Pondel: Pozyskiwanie informacji z Internetu	132
Artur Rot: Oprogramowanie dostarczane w formie usługi – model SaaS. Stan obecny, perspektywy rozwoju oraz przykłady rozwiązań	143
Jadwiga Sobieska-Karpińska, Marcin Hernes: Rozwiązywanie konfliktów w systemach rozproszonych za pomocą metod consensusu.....	154
Ryszard Zygała: Analiza modelu zarządzania efektywnością IT według Government Accountability Office.....	168

Część III. Zastosowania – narzędzia

Krzysztof Ćwikliński: The financial convergence of Warsaw and New York stock exchange in information revolution era.....	181
---	-----

Damian Dziembek: Wybrane aspekty współpracy podmiotów w ramach wirtualnego outsourcingu informatycznego	190
Karol Łopaciński: Narzędzia promocyjnej działalności organizacji w przestrzeni Internetu.....	208
Adam Nowicki, Bogdan Burkot: Usługi sieciowe jako technologia integracji systemów informatycznych wspomagających procesy biznesowe. Ocena możliwości zastosowania.....	218
Maciej Pondel: Narzędzia wyszukiwawcze w pozyskiwaniu informacji z Internetu	228
Gracja Wydmuch: Integrated platform for composite knowledge management applications. Knowledge-centric approach.....	237
Leszek Ziara: Wykorzystanie hurtowni danych we wspomaganie procesu podejmowania decyzji w przedsiębiorstwie	249

Summaries

Iwona Chomiak-Orsa: Information processes mapping as the most important stage of IT-projects	20
Wiesława Gryncewicz: The identification of information processes in Polish inland revenues	31
Dorota Jelonek: Corporate portals in the management of information resources about enterprise environment.....	41
Maja Leszczyńska: Implementing real time technologies in logistic information systems	55
Andrzej Niesler: Enterprise integration architecture with an autonomous registry of Web services	65
Monika Sitarska: Enterprise information portal as a part of knowledge and information management systems in organization.....	75
Damian Dziembek: Strategic implications for economic organizations resulting from application of virtual IT outsourcing	95
Wiesława Gryncewicz: Analysis and estimation of information quality in Polish inland revenues	109
Łukasz Łysik: The application of mobile technology in sales – the measures	120
Adam Nowicki, Mariusz Nosal: The principles of the IT governance in an enterprise	131
Jolanta Pondel, Maciej Pondel: The acquisition process of information from the Internet.....	142
Artur Rot: Software as a service (SaaS) model – current state, development perspectives and the examples of application.....	153
Jadwiga Sobieska-Karpińska, Marcin Hernes: Solving conflicts in distributed systems using consensus methods.....	167

Ryszard Zygała: An analysis of IT effectiveness management model according to IT Government Accountability Office.....	177
Krzysztof Ćwikliński: Finansowa konwergencja Giełdy Papierów Wartościowych w Warszawie i Giełdy Papierów Wartościowych w Nowym Jorku w erze informacyjnej rewolucji	188
Damian Dziembek: Chosen aspects of entities cooperation in the scope of virtual IT outsourcing.....	207
Karol Łopaciński: Instruments of organization promotional activity in Internet space.....	217
Adam Nowicki, Bogdan Burkot: Web services as the technology of business process integration. Discussing the possibilities of use.....	227
Maciej Pondel: Tools of information acquisition from Internet.....	236
Gracja Wydmuch: Zintegrowana platforma dla łącznego wykorzystania narzędzi do zarządzania wiedzą. Podejście wiedzocentryczne	247
Leszek Ziara: Data warehouses in the support of decision processes in the enterprise	254

Jadwiga Sobieska-Karpińska, Marcin Hernes

ROZWIĄZYWANIE KONFLIKTÓW W SYSTEMACH ROZPROSZONYCH ZA POMOCĄ METOD CONSENSUSU

1. Wstęp

Teoria consensusu ma swoje korzenie w teorii wyboru zajmującej się następującym problemem: dany jest zbiór Z (np. zbiór obiektów) będący podzbiorem zbioru X . Mówiąc o wyborze, dokonujemy selekcji elementów zbioru Z według pewnych kryteriów. W teorii consensusu natomiast wybór nie musi być podzbiorem zbioru Z , nie musi mieć także takiej samej struktury, jaką mają elementy zbioru Z . Początkowo teoria consensusu dotyczyła prostych struktur, takich jak porządek liniowy lub częściowy porządek. Następnie zaczęto zajmować się bardziej złożonymi strukturami, takimi jak podziały, hierarchie, n -drzewa. W pracy [Nguyen 2002] przedstawiono model consensusu pozwalający na rozwiązywanie konfliktów struktur wieloatrybutowych i wielowartościowych. Teoria consensusu dotyczy zatem problemów związanych z analizą danych w celu wydobywania użytecznych informacji (tak samo jak eksploracja danych). Cele metod eksploracji danych dotyczą poszukiwań związków przyczynowo-skutkowych ukrytych w danych, cele metod consensusu dotyczą zaś wyznaczania takiej wersji dla pewnego zestawu wersji danych, która najlepiej reprezentuje dane wersje lub jest kompromisem akceptowanym przez strony będące autorami tych wersji. Za pomocą teorii consensusu można rozwiązywać różnego rodzaju konflikty występujące na poziomie danych. W literaturze przedmiotu problemy rozwiązywane za pomocą teorii consensusu dzieli się na następujące grupy [Hernes, Nguyen 2004]:

1. Problemy związane z odkrywaniem ukrytej struktury obiektu. Na przykład: dany jest zbiór elementów, a odkrywaną strukturą jest funkcja odległości między tymi elementami.

2. Problemy związane z uzgadnianiem niespójnych lub sprzecznych danych dotyczących tego samego obiektu. Na przykład: eksperci przedstawiają różne wersje danych i należy znaleźć jedną satysfakcjonującą wersję, która zostanie przedstawiona użytkownikowi systemu.

Wyznaczanie consensusu składa się z kilku etapów. Na początku należy dokładnie zbadać strukturę zbioru Z , a następnie obliczyć odległości między poszczególnymi podzbiórami zbioru Z . Wyznaczenie consensusu polega na wyborze takiego zbioru, że odległość między tym zbiorem (consensusem) a podzbiórami zbioru Z jest minimalna (według różnych kryteriów określających, jak bardzo consensus powinien być zbliżony do poszczególnych podzbiórów zbioru Z).

Teoria consensusu wykorzystywana jest do rozwiązywania konfliktów różnych struktur danych w różnych systemach rozproszonych (opisanych w dalszej części artykułu), np. konfliktów wiedzy ekspertów, konfliktów w temporalnych bazach danych, konfliktów w systemach wieloagentowych, do przywracania spójności replikowanych danych. Istnieje wiele prac poświęconych wykorzystaniu teorii consensusu. Dotyczą one różnych struktur danych – zarówno prostych, jak i złożonych. Dla różnych struktur danych istnieją rozmaite sposoby obliczania funkcji odległości. Sposoby te zostały zaprezentowane m.in. w pracach: [Hernes 2004; Hernes, Nguyen 2004; Nguyen 2002; Sobieska-Karpińska, Hernes].

Wyniki uzyskane przez zastosowanie metod consensusu są dobrą reprezentacją danego zbioru, ponieważ biorą pod uwagę praktycznie wszystkie podzbiory rozpatrywanego zbioru, natomiast metody wyboru w dużym stopniu uwzględniają jeden z podzbiórów danego zbioru, w małym zaś stopniu – pozostałe podzbiory rozpatrywanego zbioru. Jeżeli stosujemy metody consensusu do rozwiązywania konfliktu, to consensus jest takim rozwiązaniem, które daje stronom konfliktu następujące korzyści:

1. Każda ze stron jest brana pod uwagę w consensusie.
2. Każda ze stron konfliktu „traci” najmniej jak tylko to jest możliwe.
3. Każda ze stron wnosi swój wkład w consensus.
4. Wszystkie strony akceptują consensus.
5. Consensus jest reprezentacją wszystkich stron.

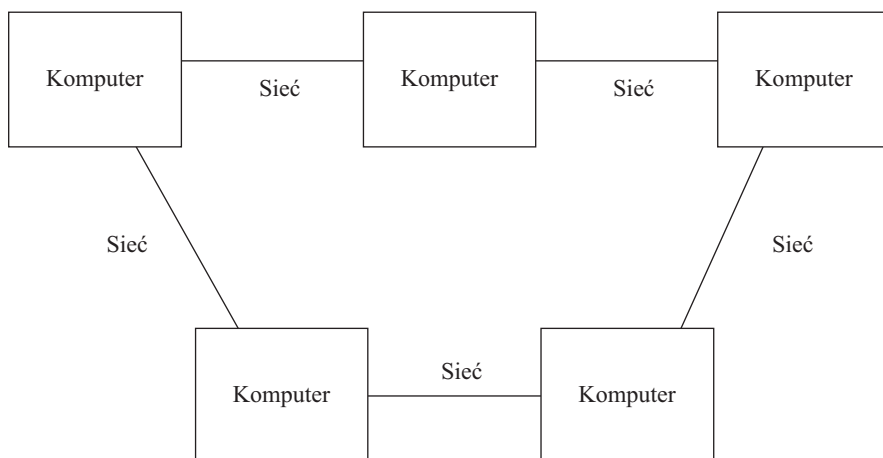
W celu przybliżenia możliwości wykorzystywania metod consensusu w niniejszym artykule zostanie dokonana ich charakterystyka i przedstawiony będzie sposób postępowania podczas wyznaczania consensusu.

2. Systemy rozproszone

Jednym z obszarów zastosowań metod consensusu są systemy rozproszone. System rozproszony (por. [Hernes 2004, s. 47; Nguyen 2002, s. 7]) jest to zbiór samodzielnych komputerów połączonych za pomocą sieci służący osiągnięciu konkretnych celów. Przykład systemu rozproszonego pokazano na rys. 1.

Podstawą działania systemu rozproszonego jest oprogramowanie, które umożliwia:

- koordynowanie działań komputerów,
- dzielenie zasobów (sprzętu, oprogramowania, danych).



Rys. 1. Przykład systemu rozproszonego

Źródło: opracowanie własne.

Należy zauważyć, że z punktu widzenia użytkownika system rozproszony powinien stanowić spójną całość, mimo że składa się z wielu urządzeń składowych (w odróżnieniu od systemów, w których użytkownik musi się samodzielnie łączyć z innymi urządzeniami np. za pomocą RS232). Systemy rozproszone są dziedziną rozwijającą się dynamicznie, znalazły one szerokie zastosowanie w procesach przetwarzania dużej ilości informacji, ponieważ mają cechy wychodzące naprzeciw oczekiwaniom współczesnych użytkowników, a więc: dzielenie zasobów, otwartość, współbieżność, skalowalność, tolerowanie uszkodzeń, przezroczystość.

Dzielenie zasobów umożliwia wykorzystywanie określonego zasobu przez wielu użytkowników systemu, co pozwala na skuteczniejsze i pełniejsze wykorzystanie części składowych systemu, a także na zmniejszenie kosztów funkcjonowania systemu. Dzielenie zasobów odbywa się na poziomie:

- sprzętu – istnieje możliwość wspólnego wykorzystywania danego procesora, pamięci twardego dysku, drukarki;
- aplikacji – te same programy mogą być wykorzystywane przez wielu użytkowników; jeśli dany program jest zainstalowany na jednym komputerze, to możemy go uruchomić na wszystkich komputerach znajdujących się w systemie bez potrzeby instalacji na każdym z osobna, co pozwala zaoszczędzić dużo miejsca na twardych dyskach;
- danych – dotyczy wspólnych plików, bazy danych; dane znajdujące się na jednym komputerze mogą być wykorzystywane przez inne komputery, co również pozwala zaoszczędzić miejsce na twardym dysku.

Jeżeli chcemy, aby dany zasób był dzielony, to musi on być zarządzany przez specjalny program zwany zarządcą zasobu, umożliwiającą dostęp do zasobu, działanie na nim, a także niezawodną i spójną aktualizację zasobu.

Otwartość to możliwość dodawania nowych usług bez naruszania lub zwielokrotniania usług istniejących [Hernes 2004]. Możemy dodawać sprzęt, oprogramowanie, dodatkowe struktury danych bez konieczności dokonywania zmian w zasobach istniejących. Na przykład w komputerach stacjonarnych mamy możliwość tylko ograniczonego rozszerzenia sprzętu, po pewnym czasie musimy wymienić całość.

System otwarty możemy konstruować z różnego rodzaju sprzętu komputerowego i oprogramowania; możemy dodawać kolejne struktury danych, a użytkownik będzie wszystko widział jako spójną całość. Przy rozszerzaniu części składowych systemu rozproszonego należy jednak pamiętać o istniejących standardach.

Otwartość systemów rozproszonych daje użytkownikowi następujące korzyści:

- możliwość zaspokojenia coraz większych potrzeb związanych z systemem;
- elastyczność systemu – użytkownik może zażądać rozszerzenia części systemu w zależności od zaistniałych warunków zewnętrznych;
- ciągłość pracy z systemem – ponieważ dodanie części składowych nie wymaga zmiany części już istniejących;
- niezależność – użytkownik, jeśli nie wyrazi chęci, nie jest zależny od modyfikacji wprowadzonych przez innych użytkowników;
- rozwój – rozszerzenie systemu daje możliwość uczestniczenia w szkoleniach dotyczących nowych elementów systemu;
- zmniejszenie kosztów związanych z systemem, ponieważ jeżeli chcemy rozszerzyć system, nie musimy zmieniać istniejących części składowych, co byłoby kosztowne.

Współbieżność pozwala na wykonywanie wielu procesów równoległe. W systemach opierających się na jednym procesorze współbieżność uzyskuje się przez dzielenie czasu procesora dla różnych procesów (proces uzyskuje dany kwant czasu, wykorzystuje procesor, po czym jest usypiany, a procesor jest przydzielany do innego procesu). W przypadku systemów rozproszonych sytuacja jest o wiele lepsza, ponieważ z definicji występuje w nim wiele komputerów, a więc procesy uruchamiane są równoległe – każdy procesor może obsługiwać inny proces.

W systemie rozproszonym istnieją dwie sytuacje, w których występują działania równoległe [Nguyen 2002]:

- wielu użytkowników jednocześnie wydaje polecenia lub współpracuje z programami użytkowymi;
- wiele procesów serwerów działa współbieżnie, obsługując klientów.

W pierwszym przypadku na każdym komputerze użytkownika uruchamiany jest jego własny proces, dzięki czemu procesor nie musi być dzielony dla procesów innych użytkowników. Użytkownik może także mieć komputer wieloprocessorowy i wtedy kilka procesów może być wykonywanych równoległe bez dzielenia czasu procesora.

W drugim przypadku każdemu zasobowi przyporządkowany jest serwer (np. dla bazy danych), na którym jest uruchomiony proces. Dzięki temu można równo-

legle wykonywać procesy serwera dla różnych zasobów. Serwery mogą być także wieloprocesorowe i wtedy mogą równocześnie obsługiwać wielu klientów bez potrzeby dzielenia czasu procesora.

Użytkownik osiąga następujące korzyści wynikające ze współbieżności:

- możliwość wykonywania wielu działań równocześnie;
- przyspieszenie wykonywania działań;
- eliminowanie konfliktów z innymi użytkownikami;
- większą pewność systemu (jeżeli nastąpi awaria komputera, to dotyczy tylko jednego użytkownika lub niewielu użytkowników).

Skalowalność oznacza możliwość rozszerzenia systemu bez konieczności wprowadzania zmian w istniejących zasobach [Nguyen 2002]. Przykładem systemu skalowalnego jest sieć Internet. Skalowalność jest cechą, na którą należy zwrócić szczególną uwagę podczas projektowania systemu. Jeżeli bowiem popełnimy błąd, może się okazać, że nie będzie możliwe rozbudowanie niektórych zasobów systemu. Na przykład: jeśli wzrośnie liczba użytkowników, którzy żądają dostępu do danych plików, to system powinien oferować możliwość dołożenia nowego serwera plików, aby nie spowodować spadku wydajności. Pojęcie wydajności jest ściśle związane ze skalowalnością systemu, gdyż często się zdarza, że rozbudowa systemu powoduje powstanie tzw. wąskich gardeł. Przykładem może być przepustowość łącza umożliwiającego dostęp do Internetu. Jeśli bowiem korzysta z niego kilku użytkowników, to prędkość transmisji jest zadowalająca, natomiast gdy z łącza korzysta kilkudziesięciu użytkowników, to prędkość może się okazać niewystarczająca.

Systemy skalowalne powinny pracować w ten sposób, aby wydajność była niezależna od rozmiarów systemu. Korzyści, jakie daje użytkownikowi skalowalność, to:

- możliwość „podłączenia się” do systemu praktycznie w każdej chwili;
- możliwość zwiększenia zasobu danego rodzaju, jeśli brakuje go w systemie;
- dostępność systemu dla coraz większej liczby użytkowników (Internet).

Przezroczystość pozwala na ukrywanie przed użytkownikiem i programistą aplikacji oddzielności składowych systemu [Nguyen 2002]. Dzięki temu użytkownik czy programista widzi system jako całość i nie musi się zastanawiać nad tym, gdzie fizycznie znajdują się aktualnie wykorzystywane zasoby.

Występowanie przezroczystości przejawia się np. w sieci Internet. Jeżeli wpisujemy żądany adres strony, to nie jest istotne, w którym miejscu na świecie fizycznie się ona znajduje. Zawsze odwołujemy się w ten sam sposób. Innym przykładem jest poczta elektroniczna. Adresat może pochodzić z różnych stron świata, natomiast sposób wysyłania listów jest jednakowy. Należy pamiętać, że przezroczystość nie zawsze jest pożądana. Jeśli np. chcemy coś wydrukować, to musimy wiedzieć, gdzie znajduje się drukarka.

Tolerowanie uszkodzeń polega na wykryciu awarii w systemie i zastosowaniu rozwiązań, które spowodują, że awaria będzie niewidoczna dla użytkownika systemu. W każdym systemie rozproszonym może dojść do awarii sprzętu czy też opro-

gramowania, jednakże wpływ uszkodzenia na pracę całości powinien być jak najmniejszy. Podczas projektowania systemu rozproszonego należy przyjąć założenie, że każda składowa systemu może ulec awarii. Przy takim podejściu niezbędne staje się stworzenie mechanizmów, które pozwolą na tolerowanie uszkodzeń.

Wymienione własności, oprócz swoich zalet, wywołują także różnego rodzaju problemy, których skuteczne rozwiązywanie jest podstawą prawidłowego działania systemu. W dalszej części artykułu zostaną omówione konflikty w systemach rozproszonych.

3. Konflikty w systemach rozproszonych

Istnieją różne definicje konfliktu; oto niektóre z nich:

1. Konflikt występuje wtedy, gdy zamierzenia obu (lub więcej) stron nie mogą zostać jednocześnie spełnione w taki sposób, w jaki chciałyby tego te strony [Sposób 2002].

2. Konfliktem nazywa się sytuację, gdy dwie niezależne jednostki mają różniące się w pewnym wspólnym temacie zdania [Hernes, Nguyen 2007; Nguyen 2002].

3. Konflikt to brak zgodności wiedzy posiadanej przez każdą ze stron konfliktu [Nguyen 2002].

Gdy powstaje konflikt, są trzy podstawowe wyjścia z takiej sytuacji [Sobieska-Karpińska, Hernes]:

1. nierozwiązanie konfliktu – cele stron konfliktu nie zostały osiągnięte.

2. rozwiązanie siłowe – każda ze stron walczy o osiągnięcie swojego celu.

Mogą tutaj wystąpić następujące skutki:

- cel został osiągnięty przez jedną ze stron konfliktu;
- żadna ze stron nie osiągnęła swojego celu;
- strony tylko częściowo osiągnęły swoje cele;
- strony osiągnęły inne cele niż zamierzały.

3. Osiągnięcie porozumienia – strony konfliktu wspólnie ustalają, jakie cele mogą osiągnąć. Wyróżnia się dwa rodzaje porozumienia:

- ustępstwo – jedna ze stron rezygnuje z osiągnięcia swojego celu, druga strona w pełni osiąga swój cel;
- kompromis – każda ze stron konfliktu modyfikuje swój cel i te zmodyfikowane cele zostają osiągnięte.

Można wyróżnić trzy główne atrybuty konfliktu [Hernes, Nguyen 2004; Nguyen 2002; Sposób 2002]:

- podmiot – określa uczestników, sprawców konfliktu;
- przedmiot – opisuje, czego dotyczy konflikt;
- treść – podaje zdania podmiotu o przedmiocie konfliktu.

Konflikty w systemach rozproszonych mogą być klasyfikowane ze względu na różne kryteria. Oto niektóre z nich:

1. Ze względu na charakter zasobu, którego konflikty dotyczą, wyróżniamy [Nguyen 2002]:

- konflikty na tle sprzętu – są to konflikty dotyczące m.in. równoczesnego dostępu do zasobów chronionych ograniczeniem pojedynczego dostępu (tzw. blokady) lub też nieprawidłowego funkcjonowania komputerów podszrywających się jeden pod drugi lub podających nieprawdziwe dane na swój temat (konflikty bizantyjskie);
- na tle oprogramowania – są to konflikty wynikające ze sposobu tworzenia oprogramowania (przezroczystości dla użytkownika, identycznego dostępu do danych lokalnych, globalnych i zdalnych oraz innych powodów);
- na tle danych – są to konflikty charakteryzujące się brakiem spójności danych, pojawiające się np. przy równoczesnym aktualizowaniu wiedzy (np. replikacje danych), uzyskiwaniu wiedzy z wielu źródeł.

2. Ze względu na przyczyny i miejsce powstania wyróżniamy [Nguyen 2002]:

- konflikty interesów węzłów – występują wtedy, gdy dwa węzły lub więcej węzłów systemu równocześnie żąda dostępu do tego samego zasobu; przykładem może być zapis do tego samego pliku;
- konflikty struktur danych – powstają w wyniku błędów w projektowaniu systemu lub niezgodności w trakcie ewolucji systemu; występują takie rodzaje konfliktów struktur danych, jak:
 - konflikty encji – występują w przypadku różnych nazw encji opisujących tę samą część świata rzeczywistego lub gdy ta sama nazwa encji opisuje różne części świata rzeczywistego,
 - konflikty atrybutów – występują w przypadku różnych nazw i dziedzin wartości atrybutów opisujących tę samą część świata rzeczywistego; system traktuje je jako odrębne atrybuty,
 - konflikty zastosowania heterogenicznych struktur danych – występują w sytuacji zastosowania różnych jednostek dla tego samego atrybutu, np. odległość w niektórych przypadkach może być podawana w kilometrach, a w innych przypadkach – w milach;
- konflikty semantyki danych – występują w sytuacji, kiedy w różnych węzłach systemu są przechowywane niespójne lub sprzeczne dane opisujące ten sam atrybut; wyróżniamy następujące rodzaje konfliktów semantyki danych:
 - jednowartościowe – występują wtedy, gdy przedmiot konfliktu stanowi jedna elementarna wartość,
 - wielowartościowe – występują wtedy, gdy przedmiot konfliktu stanowi wiele elementarnych wartości.

Rozwiązywanie wymienionych konfliktów jest bardzo istotnym elementem funkcjonowania systemów rozproszonych. Ważne jest uświadomienie sobie sytuacji konfliktowej oraz podjęcie działań zmierzających do rozwiązania konfliktu. Jednym ze sposobów rozwiązywania tych konfliktów jest stosowanie metod consensusu.

4. Metody consensusu

Teoria consensusu, jak już wcześniej wspomniano, wykorzystywana jest do rozwiązywania konfliktów różnych struktur danych w różnych systemach rozproszonych, np. konfliktów wiedzy ekspertów, konfliktów w temporalnych bazach danych, konfliktów w systemach wieloagentowych, do przywracania spójności replikowanych danych (zastosowania te będą scharakteryzowane w końcowej części artykułu). Metody consensusu możemy podzielić na [Sobieska-Karpińska, Hernes 2006]:

1. Metody konstruktywne, polegające na rozwiązywaniu problemów consensusu na dwóch poziomach: mikrostruktury oraz makrostruktury uniwersum U . Mikrostruktura zbioru U to struktura jego elementów. Makrostruktura zbioru U to jego struktura. Głównym problemem przy stosowaniu tych metod jest wyznaczenie odpowiednich funkcji odległości w taki sposób, aby uwzględniały zarówno mikrostrukturę, jak i makrostrukturę.

2. Metody optymalizacyjne, polegające na definiowaniu funkcji consensusu za pomocą reguł optymalizacyjnych. Bardzo często stosuje się w tych metodach funkcje quasi-mediany, dzięki którym consensus jest najbardziej zbliżony do wszystkich rozwiązań, z których jest wyznaczany, jednocześnie odległości consensusu do poszczególnych rozwiązań są równomierne. Zastosowanie tych metod umożliwia określenie kryteriów równomierności consensusu, które mówią, jak bardzo consensus ma być zbliżony do elementów profilu.

3. Metody wykorzystujące wnioskowanie boolowskie, polegające na zakodowaniu problemu consensusu w postaci formuły boolowskiej w taki sposób, że każdy implikant pierwszy tej formuły wyznacza rozwiązanie problemu. Wnioskowanie boolowskie jest przydatne, jeżeli liczba zmiennych i ich dziedziny nie są duże. Często metody te są wykorzystywane przy przywracaniu spójności replikowanych danych, ponieważ wtedy strukturą jest zapis binarny danych.

Należy podkreślić, że poszczególne rodzaje metod consensusu wykorzystywane są w zależności od tego, w odniesieniu do jakiej struktury danych wyznaczany jest consensus.

W dalszej części artykułu przedstawiona będzie charakterystyka formalna metod consensusu przy założeniu wieloatrybutowych i wielowartościowych struktur danych.

Przyjmijmy następujące oznaczenia:

- $\Gamma(U)$ – zbiór wszystkich niepustych podzbiorów uniwersum U ,
- $\Gamma^*(U)$ – zbiór wszystkich niepustych podzbiorów z powtórzeniami uniwersum U ,
- \cup^* – suma zbiorów z powtórzeniami.

Założmy, że struktura pewnego uniwersum U jest znana.

Zakładamy, że makrostrukturą uniwersum U jest pewna funkcja:

$$\sigma: U \times U \rightarrow [0,1], \quad (1)$$

która spełnia warunki [Nguyen 2002]:

$$a) (\forall x, y \in U) [o(x, y) \geq 0], \quad (2)$$

$$b) (\forall x, y \in U) [o(x, y) = 0 \Leftrightarrow x = y], \quad (3)$$

$$c) (\forall x, y \in U) [o(x, y) = o(y, x)]. \quad (4)$$

Funkcja o spełnia więc wszystkie warunki funkcji odległości. Zauważmy jednak, że tutaj nie zakładamy warunku nierówności trójkąta, czyli funkcja odległości nie musi być metryką. W pracy [Nguyen 2002] stwierdzono, że warunki metryczne są często nałożone na funkcje odległości, lecz w niektórych przypadkach są one zbyt mocne.

Para (U, o) jest pewną przestrzenią nazywaną przestrzenią z odległością.

Najbardziej popularnymi klasami funkcji odległości są funkcje klasy MK (minimalizująca koszty) i OU (określająca udział) scharakteryzowane szczegółowo w pracach [Hernes 2004; Hernes, Nguyen 2007; Nguyen 2002; Sobieska-Karpińska, Hernes 2006]. Funkcja odległości klasy MK między dwoma zbiorami elementów polega na określeniu minimalnego kosztu przekształcenia jednego zbioru w drugi. Funkcja odległości klasy OU między dwoma zbiorami wartości elementarnych danego atrybutu polega na określeniu udziału każdej wartości elementarnej w tej różnicy [Nguyen 2002].

Przejdźmy do zdefiniowania funkcji odległości: niech $X, X_1, X_2 \in \Gamma'(U)$, $x \in U$. W dalszej części artykułu będziemy korzystać z następujących parametrów:

$$o(x, X) = \sum_{y \in X} o(x, y), \quad (5)$$

$$o^n(x, X) = \sum_{y \in X} [o(x, y)]^n \text{ dla } n \in \mathbb{N}. \quad (6)$$

Zauważmy, że parametr (5) reprezentuje sumę odległości od elementu x uniwersum U do elementów profilu X , a wielkość (6) reprezentuje sumę n -tych potęg tych odległości. Wartość ta może być zinterpretowana jako miara równomierności odległości od elementu x do elementów profilu X . Im większa jest wartość n , tym bardziej te odległości są równomierne.

W pracy [Nguyen 2002] funkcję consensusu zdefiniowano następująco: funkcją wyboru consensusu (lub funkcją consensusu) w przestrzeni (U, o) nazywamy dowolną funkcję o postaci:

$$c: \Gamma'(U) \rightarrow \Gamma(U). \quad (7)$$

Dla profilu $X \in \Gamma'(U)$ każdy element zbioru $c(X)$ będziemy nazywać jego consensusem, natomiast cały zbiór $c(X)$ – reprezentacją profilu X . Niech C oznacza zbiór wszystkich funkcji consensusu w przestrzeni (U, o) .

Następująca definicja przedstawia aksjomaty dla funkcji consensusu [Nguyen 2002]: *niech X będzie dowolnym profilem; mówimy, że funkcja consensusu $c \in C$ spełnia postulat:*

1. *Niezawodności (Re), jeśli*

$$C(X) \neq \emptyset. \quad (8)$$

2. Spójności (Co), jeśli

$$(x \in C(x)) \Rightarrow (x \in c(X \cup \{x\})). \quad (9)$$

3. Quasi-jednomysłności (Qu), jeśli

$$(x \notin C(x)) \Rightarrow ((\exists n \in \mathbb{N}) x \in c(X \cup \{n^*x\})). \quad (10)$$

4. Proporcjonalności (Pr), jeśli

$$(X_1 \subseteq X_2 \wedge x \in c(X_1) \wedge y \in c(X_2)) \Rightarrow (o(x, X_1) \leq o(y, X_2)). \quad (11)$$

5. 1- optymalności (O_1), jeśli

$$(x \in C(x)) \Rightarrow (o(x, X) = \min_{y \in U} o(y, X)). \quad (12)$$

6. 2- optymalności (O_2), jeśli

$$(x \in C(x)) \Rightarrow (o^2(x, X) = \min_{y \in U} o^2(y, X)). \quad (13)$$

Postulaty te wyrażają pierwotne warunki dla funkcji consensusu, a więc określają różne metody consensusu.

Postulat niezawodności (8) zakłada, że dla każdego profilu zawsze można wyznaczyć consensus. Odpowiada on optymistycznemu nastawieniu: *każdy konflikt da się rozwiązać*. Niezawodność jest znanym kryterium w teorii wyboru [Daniłowicz, Nguyen 1992].

Postulat spójności (9) wymaga spełnienia warunku, że jeśli jakiś element x jest consensusem dla profilu X , to po rozszerzeniu tego profilu o x (tj. $X \cup \{x\}$) element ten powinien być consensusem dla nowego profilu. Spójność jest bardzo ważną własnością funkcji consensusu, ponieważ pozwala ona użytkownikom przewidzieć zachowanie reguł wyboru consensusu, kiedy przesłanki niezależnych wyborów są połączone ze sobą.

Według postulatu quasi-jednomysłności (10), jeśli jakiś element x nie jest consensusem dla profilu X , to powinien on być consensusem dla profilu X^1 zawierającego X i n wystąpień elementu x dla pewnego n . Innymi słowy, każdy element uniwersum U powinien być wybrany jako consensus dla takiego profilu, o ile liczba jego wystąpień jest dostatecznie duża.

Postulat proporcjonalności (11) jest dość naturalną własnością, ponieważ im większy jest profil, tym większa jest różnica między jego elementarni a consensusem wybranym dla niego.

Bardzo szczególne są dwa ostatnie postulaty. Pierwszy z nich, postulat 1- optymalności (12) wymaga, aby consensus był jak najbardziej (najbardziej podobny) elementów profilu. Postulat ten, bardzo dobrze znany w literaturze, określa konkretną klasę funkcji, zwanych medianami. Natomiast postulat 2- optymalności (13) wymaga, aby suma kwadratów odległości od consensusu do elementów profilu była najmniejsza. Powód wprowadzenia tego postulatu wynika z następującego (także bardzo naturalnego) warunku dotyczącego określenia funkcji consensusu: consensus powinien być jak najbardziej „sprawiedliwy”, co oznacza, że jego odległości do

elementów profilu powinny być jak najbardziej równomierne. Zauważmy, że liczba $o^n(x, X)$ definiowana wyżej (6) może być traktowana jako miara równomierności odległości między jakimś obiektem x a elementami profilu X . Powyższy warunek wymaga zatem, aby wartość o^n (consensus, X) była minimalna. W pracy [Nguyen 2002] pokazano, że funkcje spełniające postulat 2- optymalności są lepsze od funkcji spełniających postulat 1- optymalności z racji większej równomierności, a od innych funkcji consensusu różnią się większym podobieństwem do elementów profilu. Wynika z tego, że postulat 2- optymalności jest dobrym kryterium wyboru consensusu.

Zauważmy, że postulaty (8), (9), (10), tj. Re , Co i Qu , są niezależne od struktury uniwersum U , którą jest funkcja odległości o , podczas gdy postulaty (11), (12), (13), tj. Pr , O_1 i O_2 , są sformułowane na podstawie funkcji o . Postulaty Re , Co i Qu nadają się więc w sytuacjach, w których nie można definiować funkcji odległości (lub ogólnie: makrostruktury) dla uniwersum U .

W wielu przypadkach dobrym rozwiązaniem jest consensus najbardziej równomierny, czyli taki, który w takim samym stopniu bierze pod uwagę wszystkie możliwe rozwiązania. Wychodząc z założenia, że jeżeli postulat 2- optymalności pozwala uzyskać większą równomierność niż postulat 1- optymalności, to należałoby zdefiniować także postulat n - optymalności, który dla $n > 2$ będzie pozwalał na uzyskanie jeszcze większej równomierności consensusu niż postulat 2- optymalności. Postulat ten następująco zdefiniowano w pracy [Sposób 2002]: *mówimy, że funkcja consensusu $c \in C$ spełnia postulat n - optymalności (O_2), jeśli*

$$(x \in C(x)) \Rightarrow (o^n(x, X) = \min_{y \in U} o^n(y, X)). \quad (14)$$

Postulat ten jest uogólnieniem postulatów 1- optymalności i 2- optymalności.

W pracy [Nguyen 2002] udowodniono, że nie jest możliwe, aby funkcja consensusu spełniała jednocześnie wszystkie postulaty. Zatem szczegółowe funkcje consensusu definiowane dla różnych struktur będą się różnić w zależności od postulatów, jakie mają spełniać. W ten sposób powstaje kilka różnych metod wyznaczania consensusu opisanych wcześniej.

W dalszej części artykułu przedstawiono przykładowe zastosowania metod consensusu.

5. Przegląd zastosowań metod consensusu

W pracy [Nguyen 2002] zaprezentowano zastosowanie metod consensusu optymalizacyjnych przy rozwiązywaniu konfliktów w systemach rozproszonych temporalnych baz danych. W takich systemach często się zdarza, że danemu zdarzeniu różne węzły przyporządkowują różne momenty zajścia. Pokazano, że za pomocą metod consensusu możliwe jest integrowanie różnych opinii węzłów w celu wyznaczenia wspólnego przedziału, do którego powinien należeć czas zajścia danego zdarzenia (czyli czas właściwy zdarzenia). Czas transakcyjny zdarzenia (czas za-

pisania zdarzenia w bazie danych) może być oczywiście różny w każdym węźle. Metody optymalizacyjne zastosowano także w uzgadnianiu niespójności wiedzy w strukturach, w których występuje zależność między atrybutami [Zgrzywa 2006]. W praktyce tego rodzaju struktury występują bardzo często. Na przykład struktura charakteryzująca studenta zawiera atrybut *Kwota stypendium*, który jest zależny od atrybutu *Średnia ocen*. Wyznaczanie consensusu w przypadku zależności między atrybutami wymaga zastosowania algorytmów heurystycznych, ponieważ stanowi problem NP-zupełny.

Kolejnym przykładem jest zastosowanie metod wykorzystujących wnioskowanie boolowskie [Nguyen 202] w celu przywracania spójności replikowanych danych. Dane replikowane to te same dane przechowywane na różnych serwerach systemu. Istnieje zatem potrzeba zachowania spójności tych danych tak, aby nie występowała sytuacja, że dane na jednym serwerze mają inną postać na drugim serwerze (np. dane nie zostały uaktualnione). W celu zachowania spójności danych stosuje się protokoły sterowania metodami replikacji. Wyodrębnia się dwie grupy tych protokołów [Nguyen 2002]: optymistyczne i pesymistyczne. Protokoły optymistyczne dopuszczają możliwość chwilowego braku spójności danych, np. gdy pewne serwery nie są chwilowo dostępne (np. wystąpi awaria sieci, sprzętu). Jednak jeżeli tylko serwer staje się dostępny, dane są uaktualniane. Protokoły pesymistyczne natomiast nie pozwalają na udostępnianie użytkownikom danych niespójnych. Niestety protokoły te nie gwarantują, że po pewnym czasie działania systemu dane replikowane będą spójne. Przyczyną może być np. niewykonanie operacji zapisu na wszystkich serwerach, które tę operację wykonać powinny, zbyt duża częstotliwość zapisu danych, obciążenie łącza, wirusy, awaria serwera. Istnieje zatem niebezpieczeństwo, że na niektórych serwerach pojawią się dane niepoprawne (jeżeli nie naruszą one integralności danych, to nie zostaną zauważone przez system zarządzania bazą danych). Jeżeli zostanie stwierdzona niespójność replikowanych danych, to można oczywiście ponownie zwrócić się do źródła o udostępnienie prawidłowych danych. Jednak nie zawsze jest taka możliwość, ponieważ może się okazać, że aby odnowić nieprawidłowe dane, potrzebna jest bardzo duża liczba operacji, co może być zbyt kosztowne lub czasochłonne. Zaproponowano zatem wykorzystanie metod consensusu w celu analizy niespójnych danych. Przyjęto, że dane przechowywane są w postaci ciągów bitowych, zatem przekłamania dotyczą poszczególnych bitów. Wyznaczany jest consensus tych ciągów binarnych i ten consensus jest przyjmowany za dane prawidłowe.

Metody stosujące wnioskowanie boolowskie wykorzystywane są także w tolerowaniu uszkodzeń w systemach rozproszonych. Przyjęto, że każdy serwer wysyła ten sam komunikat (np. „czy jesteś prawidłowy”) do wszystkich serwerów. Dobre serwery odpowiedzą „tak”, natomiast serwery uszkodzone odpowiedzą „tak” lub „nie”, wcale nie odpowiedzą lub nie podszą się pod inny serwer. Na podstawie otrzymanych odpowiedzi istnieje możliwość wyznaczenia consensusu, co pozwala uzyskać prawidłowy obraz awarii w systemie.

W pracach [Daniłowicz, Nguyen 1992; Hernes 2004; Nguyen 2002; Sobieska-Karpińska, Hernes 2006] przedstawiono wykorzystanie metod konstruktywnych consensusu w uzgadnianiu niespójności wiedzy ekspertów lub agentów. Przyjęto, że wiedza ekspertów (agentów) jest reprezentowana w postaci określonych struktur, takich jak podziały uporządkowane, pokrycia uporządkowane czy też niepełne podziały uporządkowane. Struktury te są pewnym rodzajem klasyfikacji, w której jest ważna kolejność klas. Podział uporządkowany przedstawia taką sytuację, w której ekspert (agent) przyporządkowuje dany obiekt tylko do jednej klasy. Pokrycie uporządkowane natomiast pozwala przyporządkować obiekt do kilku klas lub do żadnej klasy. Niepełny podział uporządkowany występuje zaś wtedy, gdy agent (ekspert) przyporządkowuje obiekt tylko do jednej klasy lub do żadnej z klas. Pochodnymi tych struktur są ważne i hierarchiczne podziały lub pokrycia uporządkowane. W rozproszonych systemach ekspertowych lub agentowych może wystąpić różnica zdań tych agentów przejawiająca się w różnicy struktur danych lub w różnicy wartości atrybutów tych struktur. W takim przypadku użytkownik systemu nie jest pewien, którą odpowiedź agenta ma wybrać. Metody consensusu pozwalają w tym przypadku uzgodnić jedno wspólne rozwiązanie i to rozwiązanie przedstawić użytkownikowi systemu.

W pracy [Sobieska-Karpińska, Hernes] przedstawiono wykorzystanie metod konstruktywnych i optymalizacyjnych consensusu w systemach wspomagania podejmowania decyzji (*decision support systems*). Stwierdzono, że dotychczas rozwiązywanie konfliktów za pomocą metod consensusu nie znalazło zastosowania w systemach wspomagających podejmowanie decyzji. Należy jednak zauważyć, że metody te mogą się okazać bardzo przydatne, ponieważ zmniejszają do minimum ryzyko podjęcia złej decyzji, co wynika z pierwszej własności metod consensusu. Na przykład, jeśli mamy kilka rozwiązań danego problemu i stosujemy metody wyboru, nie mamy pewności, że wybierzemy rozwiązanie o żądanym poziomie satysfakcji, metody consensusu gwarantują natomiast, że wszystkie rozwiązania zostaną wzięte pod uwagę, przez co zwiększa się prawdopodobieństwo podjęcia satysfakcjonującej decyzji.

6. Zakończenie

Metody consensusu pozwalające na rozwiązywanie konfliktów w systemach rozproszonych znajdują coraz większe zastosowanie w praktyce głównie dzięki swym własnościom (przedstawionym we wstępie). Teoria consensusu jest cały czas rozbudowywana, zatem można się spodziewać jej wykorzystania w coraz nowszych dziedzinach systemów rozproszonych.

W artykule przedstawiono jedynie podstawowe informacje dotyczące metod consensusu i środowiska, w którym są wykorzystywane. Cała teoria consensusu obejmuje bowiem różne rodzaje funkcji odległości oraz rozmaite klasy funkcji consensusu.

Literatura

- Daniłowicz C., Nguyen N.T., *Metody wyboru reprezentacji podziałów i pokryć uporządkowanych*, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 1992.
- Hernes M., *Uzgadnianie niespójności wiedzy w systemach rozproszonych metodą consensusu*, [w:] *Multimedialne i sieciowe systemy informacyjne*, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2004.
- Hernes M., Nguyen N.T., *Deriving Consensus for Hierarchical Incomplete Ordered Partitions and Coverings*, "Journal of Universal Computer Science" 2007, 13(2), 317-328.
- Hernes M., Nguyen N.T., *Deriving Consensus for Incomplete Ordered Partitions*, [w:] N.T. Nguyen (red.), *Intelligent Technologies for Inconsistent Knowledge Processing*, Advanced Knowledge International, Australia 2004.
- Nguyen N.T., *Metody wyboru consensusu i ich zastosowanie w rozwiązywaniu konfliktów w systemach rozproszonych*, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2002.
- Sobieska-Karpińska J., Hernes M., *Metody consensusu w odniesieniu do hierarchicznych i ważonych niepełnych podziałów uporządkowanych*, [w:] J. Dziechciarz (red.), *Ekonometria. Zastosowania metod ilościowych*, Prace Naukowe Akademii Ekonomicznej nr 1123, AE, Wrocław 2006.
- Sobieska-Karpińska J., Hernes M., *Metody consensusu w systemach wspomagających podejmowanie decyzji* (w druku).
- Sposób G., *Rozwiązywanie konfliktów metodami consensusu w wieloagentowym systemie zbudowanym w wybranej architekturze*, praca magisterska, Politechnika Wrocławska, Wrocław 2002.
- Zgrzywa M., *Determining Consensus with Dependencies of Multi-Value Attributes Using Jaccard's Index*, [w:] *Multimedia and Network Information Systems*, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2006.

SOLVING CONFLICTS IN DISTRIBUTED SYSTEMS USING CONSENSUS METHODS

Summary

Consensus methods for solving conflicts in distributed systems are presented in this article. A characteristic of distributed systems is a different type of conflicts in this systems and some ways of solving these conflicts are presented at the first part of the article. The essence of consensus methods, the definition of distance function, which is an element of consensus, the definition of consensus function and postulates for consensus function are presented in the next part of the article. The examples of applications of consensus methods are presented at the end of the article.

Jadwiga Sobieska-Karpińska – dr hab. prof. nadzwyczajny w Katedrze Komunikacji Gospodarczej Uniwersytetu Ekonomicznego we Wrocławiu.

Marcin Hernes – mgr, słuchacz Zaocznego Studium Doktoranckiego na Wydziale Zarządzania, Informatyki i Finansów Uniwersytetu Ekonomicznego we Wrocławiu.