STUDIA INFORMATICA 2001
Volume 22 Number 1 (43)

Henryk KRAWCZYK
Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki

PODSTAWOWE KONCEPCJE ZARZĄDZANIA ROZPROSZONYMI SYSTEMAMI INFORMATYCZNYMI*

Streszczenie. Przedstawiono wybrane problemy zarządzania siecią, systemem rozproszonym i aplikacją rozproszoną. Z uwagi na popularność obiektowych modeli przetwarzania zasygnalizowano rozwiązania bazujące na architekturze OMA. Omówiono systemy: ProShere, MIMO i WOS, ilustrujące nowoczesne trendy rozwiązywania koncepcji zarządzania w środowisku intenetowym.

BASIC MANAGEMENT CONCEPTS OF DISTRIBUTED INFORMATION PROCESSING SYSTEMS

Summary. Some management problems of networks, distributed systems, and distributed applications are presented. Owing to popularity of object - oriented models solutions based on OMA architectures are discussed. Such systems as: ProShere, MIMO and WOS are discussed. They illustrate well modern trends of management concepts in the Internet environment.

1. Wprowadzenie

Rozwój sieci komputerowych, w tym wykorzystanie wydajnych węzłów i szybkich łącz transmisyjnych (Ethernet 2), nowych protokołów komunikacji (ATM, WAP), czy specjalizowanych serwerów, stwarza dalsze możliwości kreowania nowych usług i wykorzystania zaawansowanych aplikacji użytkowych [2]. Internet i skupiona wokół niego cała infrastruktura informatyczna powinna być właściwie zarządzana, co oznacza skuteczne i efektywne wykonywanie operacji z wykorzystaniem dostępnych zasobów w celu realizacji

* Pracę wykonano w ramach grantu KBN 8T11 00117.

założonego celu. Przyjmując strukturę warstwową sieci czy systemu rozproszonego [8], możemy wyróżnić trzy warstwy zarządzania przedstawione na rys. 1, które w zależności od rozpatrywanego poziomu systemu rozproszonego dzielą się na następujące kategorie zarządzania [4]:

- sieciowe - dotyczące usług komunikacyjnych odniesionych do modelu ISO/OSI dla transmisji przewodowej i bezprzewodowej oraz podstawowych komponentów sieci komputerowej, takich jak: stacje robocze, przełączniki, mosty, rutery, multipleksery, itp.,
- systemowe - obejmujące obiekty, procesy i zadania wykonujące się wewnątrz sieciowej struktury informacyjnej i współpracujące z systemem plików, rozdziałem obciążenia, monitorowaniem dostępu, adresacją (odwzorowanie DNS na IP), dokonywanie rozliczeń za usługi, czy obsługa wyjątków, itp.,
- informacyjno - aplikacyjne - związane z organizacją informacji i wykonywaniem prostych usług i aplikacji zorientowanych na zewnętrznego użytkownika (wymiana informacji, e-mail, pogawędki, przechowywanie archiwalnych dokumentów, multimedia, sprzedaż, wypożyczanie, itp.),
- usługowo - operacyjne - ukierunkowane na konkretne przedsięwzięcia użytkownika, socjalne, finansowe, techniczne, prawne, produkcyjne, prowadzące do kreowania aplikacji internetowej lub idąc dalej, nowej społeczności informacyjnej i stopniowego tworzenia miasta czy państwa cyfrowego (globalizacja).

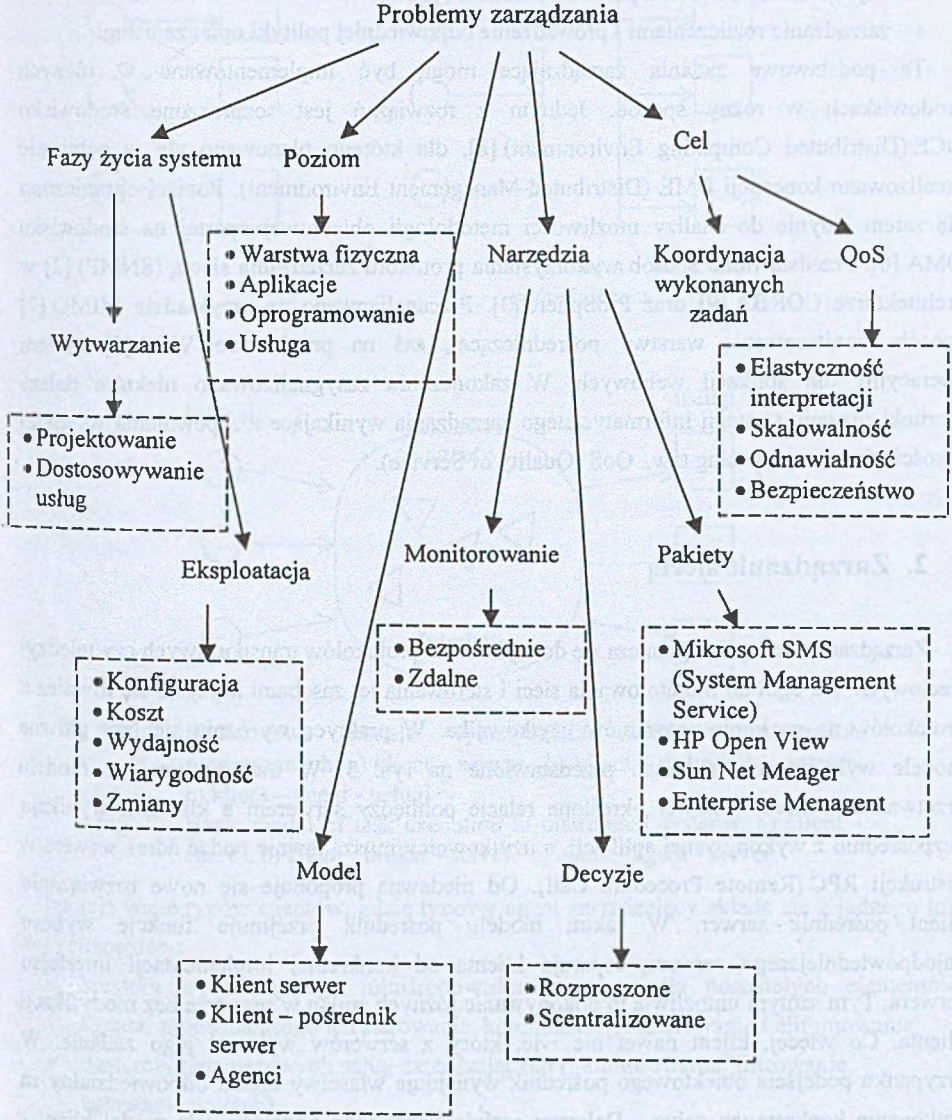
Warstwa przedsięwzięć użytkownika
Warstwa usługowo -aplikacyjna (obiekty i serwery)
Warstwa systemowo - sieciowa (komunikacja)

Rys. 1. Trzy główne warstwy zarządzania systemami informatycznymi
Fig. 1. Three basic layers of information - processing system management

Tak więc zarządzanie sieciami komputerowymi i systemami informatycznymi może być rozpatrywane z punktu widzenia wyższych kategorii, jak i innych aspektów, takich jak: rodzaj sieci (LAN, WAN, VPN, IN), typ wykorzystywanych danych (dane liczbowe, tekstowe, głosowe, obrazy, filmy), etap rozwoju usługi lub przedsięwzięcia (planowanie, instalowanie, eksploatacja, wprowadzanie zmian).

Istnieje wiele standardów precyzujących różne elementy przedstawionych strategii zarządzania. Tematyka ta jest na tyle obszerna, że poświęcono już jej kilka monografii [1, 2, 4, 8]. W sposób przybliżony oddaje to również rys. 2, gdzie przedstawiono

wybrane aspekty zarządzania systemami informatycznymi. Nie sposób je wszystkie zaprezentować w niniejszej pracy. Dlatego też skupiono się tylko nad problematyką zarządzania siecią, zarządzania systemami zbudowanymi na takiej sieci oraz zarządzania aplikacjami wykonywanymi w takim środowisku systemowym.



Rys. 2. Wybrane problemy zarządzania systemami sieciowymi
Fig. 2. Some basic problems of network system management

Ogólnie rzecz ujmując do podstawowych funkcji zarządzania należą:

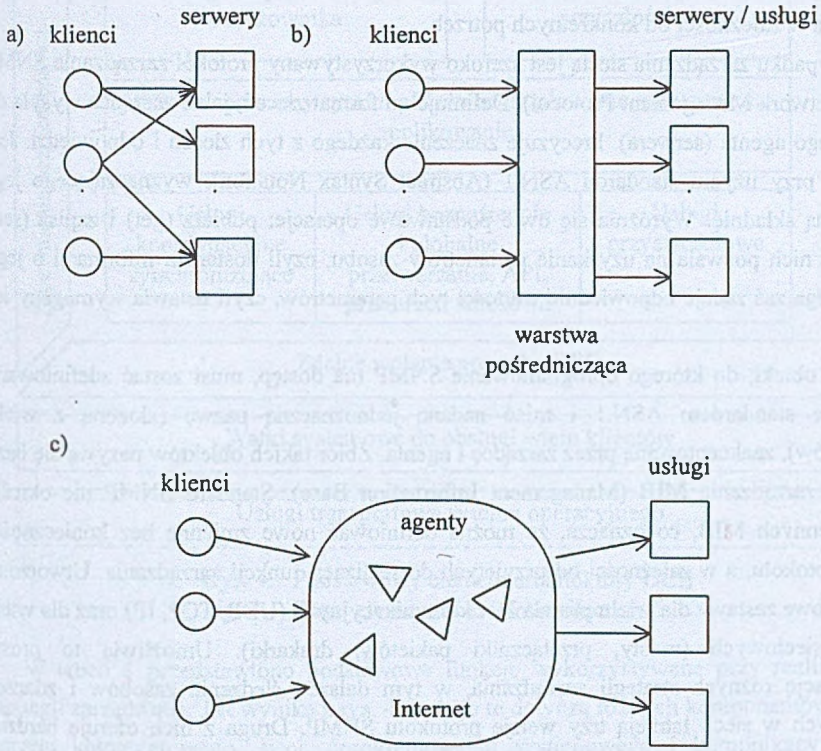
- zarządzanie konfiguracją - dobór właściwej konfiguracji do wykonywanej usługi,
- zarządzanie wydajnością - zapewnienie efektywności przetwarzanych aplikacji,
- zarządzanie wiarygodnością, w tym bezpieczeństwem - zwiększenie stopnia zaufania użytkowników, ochrona przed działalnością intruzów,
- zarządzanie rozliczeniami - prowadzenie odpowiedniej polityki opłat za usługi.

Te podstawowe zadania zarządzające mogą być implementowane w różnych środowiskach w różny sposób. Jednym z rozwiązań jest rozproszone środowisko DCE (Distributed Computing Environment) [8], dla którego planowano ale w pełni nie zrealizowano koncepcji DME (Distributed Management Environment). Poniżej ograniczono się zatem jedynie do analizy możliwości metodologii obiektowej opartej na środowisku OMA [6]. Przedstawiono sposób wykorzystania protokołu zarządzania siecią (SNMP) [2] w architekturze CORBA [9] oraz ProSphere [3]. Przeanalizowano na przykładzie MIMO [7] sposób monitorowania warstwy pośredniczącej, zaś na przykładzie WOS [5] system operacyjny dla aplikacji webowych. W zakończeniu zasygnalizowano niektóre dalsze kierunki rozwoju strategii informatycznego zarządzania wynikające z zapewnienia wysokiej jakości oferowanych usług tzw. QoS (Quality of Service).

2. Zarządzanie siecią

Zarządzanie siecią nie ogranicza się do wykonania protokołów transportowych czy międzysieciowych. Na ogół do monitorowania sieci i sterowania jej zasobami korzysta się również z protokołów na poziomie programów użytkownika. W praktyce wyróżnia się trzy główne modele wykonywania aplikacji przedstawione na rys. 3. W tradycyjnym już modelu przetwarzania klient - serwer określone relacje pomiędzy serwerem a klientem wynikają bezpośrednio z wykonywanej aplikacji, a użytkownicy muszą jawnie podać adres serwera w instrukcji RPC (Remote Procedure Call). Od niedawna proponuje się nowe rozwiązanie klient - pośrednik - serwer. W takim modelu pośrednik przejmuje funkcje wyboru najodpowiedniejszego serwera, separuje klienta od konkretnej implementacji interfejsu serwera. Tym samym umożliwia to dokonywanie różnych zmian w serwerze bez modyfikacji klienta. Co więcej, klient nawet nie wie, który z serwerów wykonał jego zadanie. W przypadku podejścia obiektowego pośrednik wynajduje właściwy obiekt odpowiedzialny za wykonanie konkretnych usług. Dalszym rozwinięciem tego podejścia jest model klient - agent - usługi, gdzie rolę pośrednika realizuje program o nazwie agent, który posiada na ogół

możliwości przemieszczania się w systemie i wykonywania wielu usług, w tym również funkcji zarządzających.



Rys. 3. Podstawowe modele wykonywania zadań w systemach rozproszonych: a) klient – serwer, b) klient – pośrednik – serwer, c) klient – agent - usługi

Fig. 3. Basic model of task execution in distributed systems: a) client - server, b) client - broker - server, c) client - agent - server

Istnieje wiele typów agentów, gdzie typowy agent zarządzający składa się z jednego lub wielu elementów:

- kreatora wytwarzającego miniśrodowisko agentowe dla pozostałych elementów agenta, zapewniającego ich sterowanie, komunikację, instalowanie i eliminowanie,
- dostarczyciela bazowych usług zarządzających (monitorowanie, filtrowanie, zgłaszanie potrzeb),
- konwertera protokołów wymaganego do zapewnienia współpracy różnych protokołów sieciowych i aplikacyjnych,

- wykonawcy konkretnych zadań zarządzających przy wykorzystaniu dostępnych usług i konwerterów protokołów.

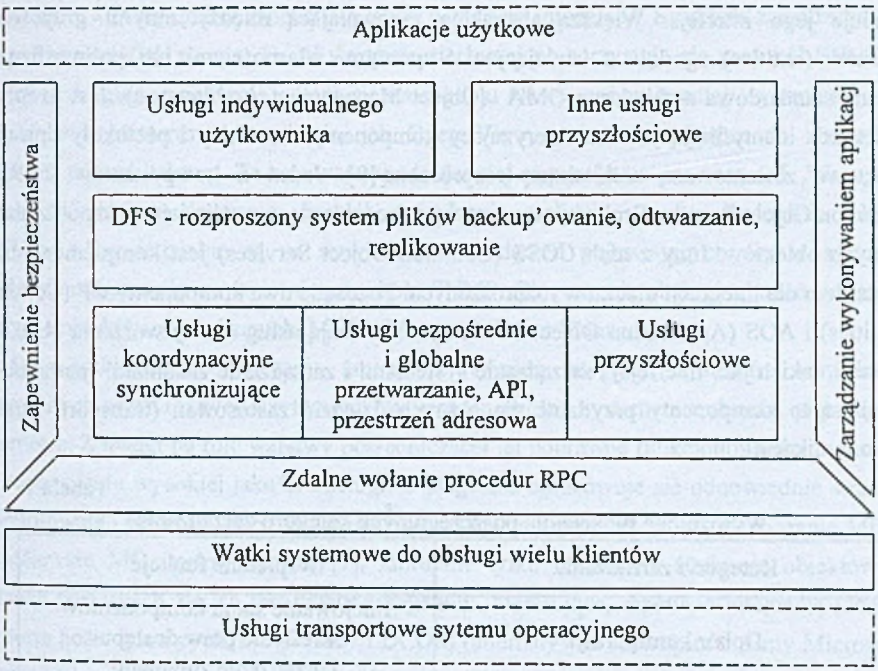
Dzięki takiej dekompozycji funkcji zarządzających możemy dobierać elastyczną strategię zarządzania w zależności od konkretnych potrzeb.

W przypadku zarządzania siecią jest szeroko wykorzystywany protokół zarządzania SNMP (Simple Network Management Protocol). Definiuje on format zleceń, jakie zarządca wysyła do niemobilnego agenta (serwera). Precyzuje znaczenie każdego z tych zleceń i odpowiedzi. Jest kodowany przy użyciu standardu ASN.1 (Abstract Syntax Notation), wyznaczającego jego abstrakcyjną składnię. Wyróżnia się dwie podstawowe operacje: pobierz (get) i zapisz (set). Pierwsza z nich pozwala na uzyskanie parametrów zasobu, czyli dostarcza informacji o jego stanie, druga zaś zadaje odpowiednie wartości tych parametrów, czyli ustawia wymagany ich stan.

Każdy obiekt, do którego oprogramowanie SNMP ma dostęp, musi zostać zdefiniowany zgodnie ze standardem ASN.1 i mieć nadaną jednoznaczną nazwę (złożoną z wielu przedrostków), zaakceptowaną przez zarządcę i agenta. Zbiór takich obiektów nazywa się bazą informacji zarządzania MIB (Management Information Base). Standard SNMP nie określa stanu zmiennych MIB, co oznacza, że można definiować nowe zmienne bez konieczności zmiany protokołu, a w zależności od przyjętych do realizacji funkcji zarządzania. Utworzono jednak bazowe zestawy dla wielu protokołów komunikacyjnych (UDP, TCP, IP) oraz dla wielu zasobów sieciowych (mosty, przełączniki pakietów, drukarki). Umożliwia to prostą implementację różnych strategii zarządzania, w tym dalsze śledzenie zasobów i zdarzeń zachodzących w sieci. Istnieją trzy wersje protokołu SNMP. Druga z nich oferuje bardziej bezpieczną komunikację, trzecia zaś jest zorientowana już na środowisko Internet.

3. Zarządzanie systemami

Punktem wyjściowym do zaprezentowania funkcji zarządzających jest konkretny system rozproszony realizowany w oparciu o sieć komputerową. Reprezentatywną propozycją jest środowisko DCE, które staje się szeroko akceptowalnym standardem. Warstwową architekturę podstawowego komponentu DCE przedstawia rys. 4. Składa się on z wielu warstw, poczynając od mechanizmów RPC, poprzez metody przetwarzania i koordynacji czasu obliczeń, systemu plików i różnego typu usług skierowanych na użytkowników. Warstwy te bezpośrednio współpracują z mechanizmami zarządzania wykonaniem aplikacji oraz zapewnienia bezpieczeństwa całego systemu.



Rys. 4. Podstawowy element architektury DCE
Fig. 4. Basic element of DCE architecture

W tabeli 1 przedstawiono podstawowe funkcje wykorzystywane przy realizacji danej strategii zarządzania. Jak wynika z rys. 4, funkcje te dotyczą różnych komponentów systemu i sprzętu komputerowego, sieci, oprogramowania systemowego i komunikacyjnego oraz aplikacji. DCE w standardzie zapewnia już pewne podstawowe funkcje, takie jak weryfikacja autentyczności użytkowników, zdalne wywoływanie procedur, wspólne używanie plików rozproszonych i zarządzanie konfiguracjami. System weryfikacji autentyczności użytkowników jest bardzo podobny do systemu Kerberos, zaś funkcje wspólnego używania plików są bardzo podobne do Andrew File System.

Bezpieczeństwo systemu DCE opiera się na programie Security Serwer, w którym otrzymywana jest lista kontroli dostępu do różnych operacji i każdorazowo przy odwołaniu się użytkownika sprawdza się, czy ten klient ma prawo do żądanej operacji. Klienci komunikują się z serwerami za pomocą Autenticated RPC. Użycie tej funkcji przez klienta lub usługi wymaga tajnego klucza, który jest znany tylko używającemu i serwerowi bezpieczeństwa (Security Serwer).

Mechanizmy RPC wykorzystywane w środowisku DCE wspierają enkapsulacje w zakresie indywidualnego serwera. Tak więc serwer akceptuje tylko te operacje, które

definiuje jego interfejs. Większą abstrakcję zapewniającą między innymi grupowanie obiektów (entities) zgodnie z istniejącymi wspólnymi własnościami lub polimorfizmem posiada standardowa architektura OMA (Object Management Architecture). Jest to model odniesienia identyfikujący i charakteryzujący komponenty, interfejsy i protokoły opisujące obiektowo zorientowaną architekturę rozproszoną [9]. Jeden z komponentów CORBA (Common Object Request Broker) jest warstwą pośredniczącą, umożliwiającą współdziałanie zdalnych obiektów. Inny z nich COSS (Common Object Services) jest komplementarnym standardem dla integracji obiektów rozproszonych. Pozostałe dwa komponenty CF (Common Facilities) i AOS (Application Object Services) dostarczają usług do wytworzenia obiektów aplikacji, takich jak: interfejsy, zarządzanie systemami i zarządzanie zadaniami (patrz tabela 1) lub są to komponenty przydatne dla różnych dziedzin zastosowań (transport, handel, telekomunikacja).

Tabela 1

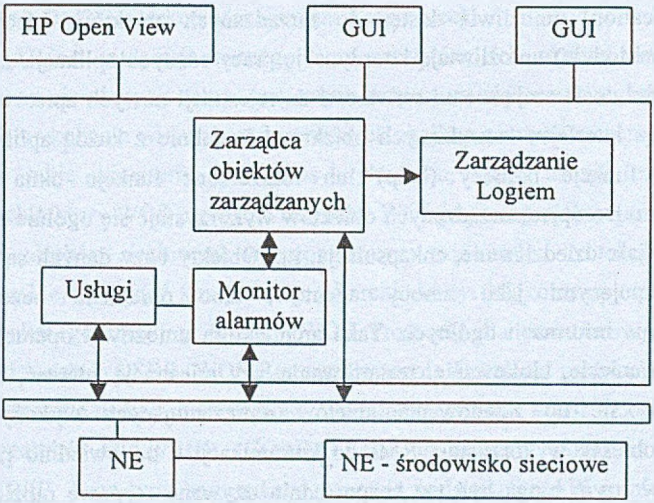
Wyróżnione funkcje dla poszczególnych kategorii zarządzania

Kategorie zarządzania	Wspierane funkcje
Dobór konfiguracji	Inicjowanie stanu komponentów Określenie praw dostępu Zarządzanie zmianami
Zapewnienie wydajności	Monitorowanie wydajności Analiza wydajnościowa Strojenie systemu
Obsługa błędów	Detekcja błędu Identyfikacja defektu Eliminacja skutków
Zapewnienie bezpieczeństwa (zabezpieczenia)	Dekodowanie informacji Zapewnienie autentyczności Obsługa alarmów Przestrzeganie praw dostępu Śledzenie zachowań systemu
Dokonywanie rozliczeń	Rejestracja usług i wykorzystywanych zasobów

Bardzo interesującym rozwiązaniem zarządzania systemami jest architektura ProSphere. Wykorzystuje ona zbiór serwerów CORBy jako podstawowe funkcje zarządzania i aplikacje Javy do prezentacji informacji dostarczonej przez te serwery. Taka architektura jest łatwo rozbudowywana, co umożliwia dołączanie nowych zasobów sieciowych bez istotnej rozbudowy oprogramowania, jest otwarta dzięki interfejsom definiowanym w języku IDL (Interface Definition Language), pozwala na dołączenie się nowych użytkowników i jest przenośna, gdyż dzięki Javie programy klienckie mogą być wykonywane na różnych

maszynach, a przeglądarki zapewniają łatwy dostęp do informacji. Struktura zarządzania jest trójpoziomowa (jak na rys. 1). W warstwie sieciowej wykorzystuje się protokół SNMP, w warstwie drugiej zarządzanie opiera się na modelu OMA, zaś w trzeciej wykorzystuje się standardowe platformy zarządzania, jak Open View, będące interfejsem do wszystkich obiektów tej architektury. Te trzy warstwy dokładniej są przedstawione na rys. 5. Jak wynika z tego rysunku, podstawowe funkcje zarządzania realizuje warstwa pośrednia. Usługi CORB'y odpowiedzialne są za serwis nazw i zdarzeń pojawiających się w systemie. Śledzenie odbywa się na poziomie protokołu SNMP. Główną rolę obsługi zdarzeń przejmuje zarządca obiektów zarządzanych MOM (Managed Object Manager).

Taka architektura zarządzania umożliwia również integrację systemów zarządzania budowanych przez różnych użytkowników w jeden ogólny system zarządzania, pracujący w Internecie. Z uwagi na rolę warstwy pośredniczącej jej poprawne funkcjonowanie jest istotne do zapewnienia wysokiej jakości obsługi. Dlatego też opracowuje się odpowiednie strategie monitorowania pracy tej warstwy. Jedną z interesujących propozycji jest rozwiązanie MIMO (Middleware MOnitoring System) [7], które nie tylko umożliwia śledzenie obiektów na różnych poziomach ale, co jest przede wszystkim interesujące, zapewnia integrację różnych platform pośredniczących, np. CORBy i DCOM (alternatywne rozwiązanie firmy Microsoft). Poza tym jest to infrastruktura wspierająca rozwój aplikacji zarządzających w całym cyklu życia, dostarczająca przyjaznego interfejsu i pozwalająca na instrumentacje śledzonych obiektów.



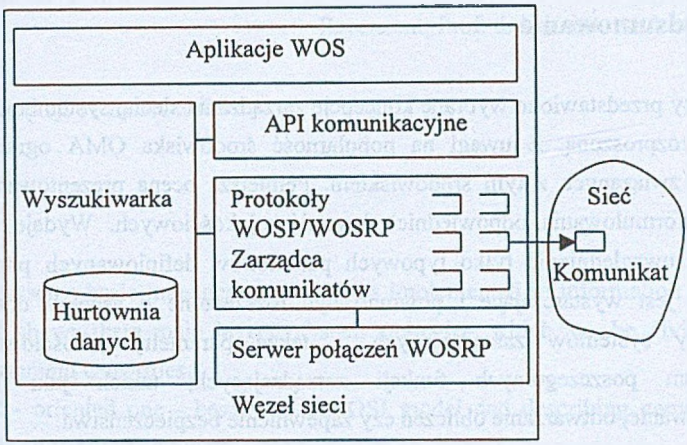
Rys. 5. Architektura ProSphere
Fig. 5. ProSphere architecture

4. Zarządzanie aplikacjami webowymi

Natura aplikacji WWW wpływa również na strategie zarządzania. Jeżeli przeglądarki wykorzystują konsole zarządzające, a zasoby są określane poprzez serwery, to dostęp do aplikacji jest możliwy z dowolnego węzła i wówczas HTTP może być stosowany jako protokół zarządzania. Jest to tanie rozwiązanie, które nazywa się wbudowanym zarządzaniem webowym. Alternatywnym podejściem jest zarządzanie z wykorzystaniem serwera proxy, gdzie komunikacja z zasobami odbywa się poprzez protokół SNMP lub CIMP, zaś protokół HTTP jest stosowany tylko jako interfejs. Takie rozwiązanie bazuje też na rozwiązaniach standardowych i może być stosowane, gdy przetwarzane są duże zbiory danych. Możliwa jest także strategia mieszana. Spośród różnych implementacji tych koncepcji przedstawiono trzy propozycje dotyczące:

1 - grupy związanej z firmą Sun - Java Management API Architecture (JMAPIA),
2 - grupy związanej z firmą Microsoft - Web Based Enterprise Management Initiative (WBEM),
3 - grupy WOS - Web Operating System (WOS).

Koncepcja JMAPIA wykorzystuje zarządzanie typu proxy i jej nowoczesną wersją jest Java Management Kit 3.0. Przeglądarka odgrywa rolę klienta, zaś platforma (serwer zarządzanych obiektów) rolę serwera. Wykorzystuje się przy tym dwa typy agentów (SNMP oraz zarządzające agenty wykorzystujące metody rodzime Javy). Mechanizm RMI (Remonte Method Invocation) umożliwia dostęp do zarządzanych obiektów. Mechanizmy AVM (Admin View Module) umożliwiają koordynację pracy różnych aplikacji, np. odpowiednie uchwyty klienta dostarczają metod serwerowi do rejestracji nowych apletów i stron HTML lub rozbudowy interfejsu zarządzanych obiektów. Podobnie z każdą aplikacją mogą być zintegrowane funkcje pomocy (help) lub rozszerzone funkcje okna współpracy z użytkownikiem. Do opisu zarządzanych obiektów wykorzystuje się ogólnie znane własności obiektowości, jak: dziedziczenie, enkapsulacja, itp. Obiekty bazy danych są obiektami typu proxy, występującymi jako zasoby agentów, albo obiektami bazodanowymi do przechowywania informacji ogólnych. Taka architektura umożliwia operacje na obiektach (kreowanie, usunięcie, blokowanie, restartowanie, czy rejestrację zdarzeń, itp.). Natomiast HTTP stosuje się do załadowania apletów reprezentujących wykonywane funkcje. Zarządzane obiekty wykorzystują RMI do komunikacji z odpowiednio przygotowanymi agentami, przy czym mogą być też bezpośrednio używane typowe możliwości agentów SNMP.



Rys. 6. Architektura systemu operacyjnego dla aplikacji webowych
Fig. 6. Architecture of Web operation system

Koncepcja WBEM skupiła się nad aspektami zarządzania informacją. Został opracowany model CIM (Common Information Model) precyzyjnie opisujący wykorzystywane w tym celu informacje. Określono też budowę zarządzanych obiektów MOF (Management Object Format). Dokonano też specyfikacji CIM w języku XML. Odwzorowując CIM w XML umożliwia się przeglądarce wyświetlanie informacji sterujących. Bazując na takim podejściu dokonano definicji dwóch platform dla specjalizowanych zastosowań: HMM (Hypermedia Management) oraz DMTF (Deskop Management Task Force).

Ideą WOS [5] jest integracja różnego typu serwerów, ich różnych wersji, a także oferowanych przez nie różnych usług, przy czym dotyczy to złożonych żądań użytkownika, które wymagają wykonania wielu skoordynowanych akcji na różnych sewerach. Dodatkowo uwzględnia się takie parametry, jak: czas wykonania, koszt usługi, czy jej jakość. Komponenty architektury WOS przedstawia rys. 6. WOS jest tak projektowany, by zapewnić przeźroczystość zasobów sieciowych w rozproszonym środowisku. Protokoły komunikacji stanowią niejako klej spajający komponenty tej architektury. Wykorzystuje się dwa główne protokoły: protokół generacyjny WOSP (WOS Protocol) dla podstawowych usług oraz protokół obsługi żądań WOSRP (WOS Request Protocol). Instalacja protokołu WOSP może być zaimplementowana na wyższym poziomie jako interfejs XML - CBL (Common Business Library) lub protokół GIOP/IIOP (CORBA), zaś na niższym poziomie jako rodzina protokołów TCP/IP. Tego typu komunikaty stanowią podstawowe API implementowane w języku Java i wykorzystujące mechanizm RMI. Jest też możliwe zintegrowanie WOS z CORBą [6], co oznacza, że klienci WOS mogą wykorzystać usługi CORBy i vice versa.

5. Podsumowanie

W pracy przedstawiono wybrane koncepcje zarządzania siecią, systemem rozproszonym i aplikacją rozproszoną. Z uwagi na popularność środowiska OMA ograniczono się do rozwiązań związanych z tym środowiskiem. Pełniejsza ocena prezentowanych rozwiązań wymaga sformułowania odpowiednich kryteriów jakościowych. Wydaje się, że w tym przypadku uwzględnienie tylko typowych parametrów definiowanych poprzez standardy (QoS) nie jest wystarczające i powinno być rozszerzone o aspekty dotyczące jakości architektury systemów zarządzających, a także parametry jakościowe związane z wykonaniem poszczególnych funkcji zarządzających, takich jak: monitorowanie, administrowanie, odtwarzanie obliczeń czy zapewnienie bezpieczeństwa.

LITERATURA

1. Amatiza B., Asheen C., Unni W.: Network Management of TCP/IP Networks: Present and Future. IEEE Network Magazine, July 1990, str. 35-43.
2. Conner D.E.: Sieci komputerowe i intersieci. WNT, Warszawa 2000.
3. Haggerty P., Seetharamoen K.: The Benefits of CORBA - Based Network Management. Communication of the ACM, vol. 4, No 10, str. 73-79.
4. Hegering H.G., Abeck S., Neumair B.: Integrated Management of Networked Systems. Morgan Kaufman Publishers, San Francisco 1998.
5. Kropt P.: Overview of the WOS project. Materiały Konferencji Advanced Simulation Technologies ASTC, San Diego, CA, USA, kwiecień 1999.
6. Krone O., Josef A.: Using CORBA in the Web Operating Systems. Lecture Notes in Computer Science. No. 1830 DCW 2000, Springer-Verlag, Berlin 2000, str. 133-141.
7. Rockl G.i inni: MIMO - An Infrastructure for Monitoring and Managing Distributed Middleware Environments. Lecture Notes in Computer Science. Nr 1795, Middleware, Springer-Verlag, Berlin 2000, str. 71-87.
8. Tanenbaun A.S.: Rozproszone systemy operacyjne. PWN, Warszawa 1997.
9. Widmer B., Lugmayer W.: A Comparison of Three CORBA Management Tools. W książce U. Emmericha i V. Gruhna pt. Engineering Distributed Objects (EDO'99), Los Angeles, May, 1999, str. 12-21.

Recenzent: Prof. dr hab. inż. Andrzej Grzywak

Wpłynęło do Redakcji 31 marca 2001 r.

Abstract

The paper presents basic management strategies implemented in information processing systems. Fig. 1 shows there main layers of such strategies, which can be divided in the following management categories:

1. network - oriented one - based on ISO/OSI model and describing communication protocols and network devices,
2. system - oriented one - describing objects, processes and tasks executed inside a network infrastructure,
3. application - oriented one - defining simple services and applications offered by a system to extend users,
4. enterprise - oriented one - connected with as human activities as business, low, production, transportation and design.

Fig 2 summarises main problems of these categories. Besides, three main layers are presented in details. Therefore SNMP protocols used for network management are described and their utilisation in different models of system application is shown. Fig. 3 defines three models of task execution: classical client - sever, client - broker - server and client - agent - services. All these models allows to define various management strategies, which can be implemented in different computing environments. We limit our consideration only to object - oriented methodology. In the paper DCE architecture is considered (see Fig. 4) and main components responsible for execution of management tasks are described. Moreover, very interesting ProSphere architecture (Fig. 5) is also given. It allows to integrate wide available commercial solutions, such as HP Open View with OMA architecture and Internet. Because the middleware plays essential role in all management strategies, a concrete example of middleware monitoring system (MIMO) is also analysed.

In order to demonstrate management idea of web applications three solutions are given: Java Management API Architecture, Web Based Enterprise Management and Web Operating System. They facilitate controlling and monitoring of an application behaviour by typical browsers. Besides, WOS architecture (see Fig. 6) allows to co-ordinate execution of user requests occupying simultaneously many servers working under CORBA facilities.