ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1995

Seria: INFORMATYKA z. 28 Nr kol. 1270

Jakub SZYMASZEK
Krzysztof ZIELIŃSKI

WYBRANE ASPEKTY STOSOWANIA SYSTEMU ANSA W ROZPROSZONYCH APLIKACJACH MULTIMEDIALNYCH [1]

Streszczenie. Istniejące środowiska programowania rozproszonego, takie jak ANSA, nie spełniają wymagań aplikacji multimedialnych. W artykule omówiono koncepcję rozszerzeń systemu ANSA, które umożliwiłyby łatwe tworzenie takiego oprogramowania. Następnie przedstawiono architekturę systemu komunikacji głosowej zrealizowanego w oparciu o podstawową wersję środowiska ANSA.

SELECTED ASPECTS OF USING ANSA SYSTEM FOR DISTRIBUTED MULTIMEDIA APPLICATIONS

Summary. Most existing environments for distributed programming, such as ANSA, do not comply with requirements of multimedia systems. In this paper, concepts of ANSA system extensions enabling easy multimedia applications programming are presented. Futhermore, an architecture of voice communication system based on basic version of ANSA environment is described.

[1]Pracę wykonano w ramach grantu KBN nr 8 S503 015 06.

ASPECTS D'UTILISATION DE SYSTÈME ANSA POUR LES APPLICATIONS MULTIMÉDIALES REPARTIS

Résumé. La majorité d'environements de programmation reparti, comme ANSA, n'est pas compatible avec les demandes des applications multimédiales. Dans l'article les ameliorations de cettes systémes sont proposé est verifiée sur l'example d'une application de transmission de la parole.

1. Wprowadzenie

Zagadnienia tworzenia oprogramowania dla multimedialnych systemów rozproszonych znajdują się w centrum uwagi licznych prac badawczych [6, 8, 9]. Nowe media informacji, takie jak głos oraz ruchomy obraz, wnoszą bowiem szereg wymagań odnośnie do trybu ich przetwarzania, które muszą być uwzględnione w przyjętym modelu obliczeniowym, funkcjach systemowych i konstrukcjach językowych.

Celem artykułu jest przedstawienie tych wymagań w kontekście otwartych systemów programowania rozproszonego. Wykorzystano w tym celu rozważania [7] dotyczące rozszerzeń systemu ANSA (Advanced Networked Systems Architecture) [1, 2, 3, 4, 5] w kierunku stworzenia architektury oprogramowania dla systemów multimedialnych. Elementy tej propozycji uwzględniono w implementacji systemu komunikacji głosowej w sieci Ethernet. Wykazano w ten sposób przydatność bazowego modelu ANSA do implementacji rozproszonych systemów multimedialnych. Przyjęte rozwiązania stanowią pewien przypadek szczególny implementacji modułów systemowych, proponowanych jako rozszerzenia modelu bazowego.

Konstrukcja artykułu jest następująca. W sekcji 2 przedstawiono ogólną analizę wymagań stawianych multimedialnym systemom rozproszonym. W następnej sekcji pokazano sposób spełnienia tych wymagań poprzez wskazanie rozszerzenia bazowego systemu ANSA o określone nowe elementy architektury. Sekcja 4 zawiera koncepcję użycia bazowego systemu ANSA do implementacji systemu komunikacji głosowej w sieci Ethernet. Sekcja 5 omawia szczegóły implementacji poszczególnych składowych systemu w odniesieniu do ogólnych rozważań prowadzonych w sekcji 2 i 3.

2. Wymagania multimedialnych systemów rozproszonych

Większość istniejących środowisk programowania rozproszonego nie spełnia wymagań systemów multimedialnych, takich jak duże prędkości transmisji i przetwarzanie w czasie rzeczywistym wielu strumieni danych jednocześnie. Analizę powstających problemów celowo jest przeprowadzić w odniesieniu do pięciu elementów składowych multimedialnych systemów rozproszonych:

- media przekazywania informacji,
- podsystem komunikacji,
- infrastruktura rozproszonego przetwarzania informacji,
- aplikacje,
- interfejs użytkownika.

W tym punkcie zostaną rozważone tylko pierwsze trzy składowe. Pozostałe zostaną przeanalizowane nieco szerzej w kolejnych punktach.

Ciagłe media, takie jak głos, a w szczególności obraz, wymagają znacznie większych szybkości transmisji informacji niż te, które są powszechnie dostępne w istniejących systemach komputerowych. Co prawda, nowoczesne metody kompresji informacji zmniejszają to wymaganie odnośnie do pojedynczego strumienia danych, to jednak konieczność utrzymywania wielu jednoczesnych połączeń sprawia, iż potrzebna zagregowana szybkość jest nadal bardzo duża. Współczesne stacje robocze są nadal w przeważającej większości jednoprocesorowe i szyna komunikacji z pamięcią stanowi ich wąskie gardło. Jej wysoki stopień wykorzystania nie gwarantuje dla zewnętrznego strumienia danych potrzebnych parametrów transmisji. Problem ten próbuje się rozwiązać przenosząc zadanie przetwarzania informacji multimedialnej do urządzeń peryferyjnych. Sprawia to, że zarządzanie tymi urządzeniami staje się złożone i wpływa istotnie na parametry systemu.

Istniejące sieci komputerowe, takie jak Ethernet czy FDDI, nie spełniają wymagań odnośnie do wysokiej gwarantowanej przepustowości przy jednoczesnym zachowaniu małego i stałego opóźnienia (*latency*). Dopiero sieci typu ATM spełniają to wymaganie na poziomie fizycznym. Niestety aktualna implementacja protokołów komunikacyjnych wyższych warstw (np. IP, TCP, UDP) wnosi znaczące opóźnienia, widoczne zwłaszcza przy transmisji tam i z powrotem pomiędzy dwoma procesami. Sprawia to, że cały system komunikacyjny nadal nie posiada wymaganych parametrów. Aby poprawić tę systuację, opracowuje się nowe implementacje istniejących protokołów oraz projektuje nowe. Wysiłki projektantów koncentrują się na minimalizacji liczby kopiowań informacji oraz liczby poziomów, na których realizowana jest multipleksacja strumieni danych.

Trudności związane z rozproszonym przetwarzaniem informacji są związane z dwoma podstawowymi zagadnieniami: heterogenicznością sprzętu i protokołów komunikacji oraz nieodpowiednim zarządzaniem zasobami. Heterogeniczność implikuje konieczność transformacji informacji, co wprowadza dodatkowy narzut i opóźnienia. W zakresie zarządzania zasobami systemy operacyjne ogólnego przeznaczenia konstruowane są raczej przy założeniu sprawiedliwego dostępu wszystkich procesów do procesora, buforów, sieci itp., niż gwarantowanych parametrów obsługi, typowych dla działania w czasie rzeczywistym. Zastosowanie systemów operacyjnych czasu rzeczywistego nie stanowi rozwiązania problemu, gdyż środowisko programowania, jakie dostarczają, jest zazwyczaj zbyt ubogie.

Przedstawiona analiza zawiera ogólne postulaty, które powinny być uwzględnione między innymi w celu przystosowania istniejących środowisk programowania rozproszonego do przetwarzania informacji multimedialnej. W szczególności zostanie to pokazane w odniesieniu do systemu ANSA poprzez omówienie kierunków rozszerzeń tego systemu ważnych dla tej klasy zastosowań.

3. Rozszerzenia bazowej architektury ANSA

Aplikacja rozproszona może być opisana za pomocą jednego z pięciu modeli środowiska ANSA, co w rezultacie daje pięć różnych punktów widzenia na system rozproszony. Najważniejsze z nich to model obliczeniowy i inżynieryjny. W modelu obliczeniowym podstawowym pojęciem jest usługa (*service*). Komponenty aplikacji — obiekty obliczeniowe (*computational objects*) — wykonujące usługi, nazywane są serwerami, a komponenty korzystające z usług — klientami. Serwery udostępniają klientom wykonywane przez siebie operacje za pomocą interfejsów. Kojarzenie ofert serwerów z żądaniami klientów następuje w *Traderze* — standardowym obiekcie obliczeniowym systemu ANSA. Dostępne usługi są pamiętane w specjalnej bazie danych *Tradera*, w której są rejestrowane w wyniku operacji *export* wykonanej przez implementujący je serwer. Przed użyciem klient musi dokonać *importu* usługi z bazy danych.

Po kompilacji elementy modelu obliczeniowego stają się elementami modelu inżynieryjnego, które mają strukturę hierarchiczną. Jednostkami najwyższego rzędu są kapsuły (*capsules*) składające się z obiektów, które z kolei mogą posiadać dowolną liczbę interfejsów (reprezentujących usługi) udostępniających na zewnątrz operacje. Wyróżniamy dwa rodzaje kapsuł: zwykłe i zarządzane (*managed*). Kapsuły zwykłe mogą być tworzone bezpośrednio przez użytkownika. Po utworzeniu kapsuły wykonywana jest w niej zdefiniowana przez programistę procedura *body*. Jeżeli kapsuła pełni rolę serwera, to po wykonaniu procedury *body* przechodzi w stan oczekiwania na zlecenia klientów. Kapsuły zwykłe nie zawierają obiektów, lecz tylko interfejsy i operacje. Kapsuły zarządzane są tworzone i usuwane przez inne kapsuły danej aplikacji za pośrednictwem specjalnej usługi środowiska ANSA — *Factory*. Kapsuły aplikacji mogą również tworzyć i usuwać obiekty wewnątrz kapsuł zarządzanych. Aplikacja najczęściej składa się z wielu komunikujących się ze sobą kapsuł reprezentujących oddzielne przestrzenie adresowe (procesy systemu operacyjnego), które mogą znajdować się na różnych maszynach (nazywanych węzłami — *nodes*). Zasobami każdego węzła zarządza specjalny obiekt obliczeniowy — *Node Manager*.

Podana charakterystyka systemu ANSA wystarcza na obecnym etapie do sformułowania w sensie ogólnym jego rozszerzeń potrzebnych dla zastosowań multimedialnych. Przyjęto przy tym założenie, aby proponowany zbiór rozszerzeń był minimalny. Obejmuje on następujące pojęcia:

- *Strumień* — pojęcie reprezentujące pojedyncze ciągłe medium informacji oraz akt komunikacji. Reprezentacja ta uwzględnia bezpośrednio mechanizmy synchronizacji wielu powiązanych strumieni oraz możliwość synchronizacji wykonania aplikacji ze zdarzeniami strumienia.

- *Typ strumienia* — z punktu widzenia programowania strumień jest jednokierunkowym, połączeniowo zorientowanym przepływem informacji pomiędzy źródłem i ujściem, które stanowią końce strumienia. Mają one określony w kodzie programu, przypisany im typ i mogą być łączone tylko z uwzględnieniem relacji zgodności typów.

- *Urządzenie* — stanowi rozszerzenie pojęcia usługi w architekturze ANSA o możliwość powiązania operacji interfejsu z końcami strumieni. Pozwala to na reprezentacje w formie interfejsów programowych, takich urządzeń jak kamery, mikrofony, głośniki czy pamięci obrazu. Urządzenia są odpowiedzialne za generację, komunikację i reprezentację danego medium informacyjnego.

- *Typ urządzenia* — podobnie jak strumieniowi również urządzeniu przypisany jest określony typ. Typ urządzenia jest kombinacją bazowych usług ANSA, np. operacji oraz typów strumieni z nim związanych. Dzięki zdefiniowaniu relacji zgodności typów dla urządzeń mogą one być w podobny sposób zarządzane przez *Trader* jak interfejsy.

- *Jakość usługi* — z każdą operacją można związać zbiór opcji definiujących możliwe warianty jakości jej wykonania QoS (*Quality of Service*). W czasie wywołania operacji klient może zażądać dla niej odpowiedniego QoS. Żądanie to posiada charakter więzów nałożonych na zbiór opcji QoS implementacji operacji.

- *Koordynacja* (*Orchestration*) — zbiór funkcji zarządzania wymaganych do koordynacji strumieni, urządzeń, QoS oraz usług.

Zdefiniowane pojęcia są związane z wprowadzeniem następujących nowych usług systemowych w architekurze ANSA:

QoS Manager — systemowa baza danych dostępnych opcji QoS.

User Locator — moduł odpowiedzialny za lokalizacje danego użytkownika. Do modułu tego odwołują się aplikacje, które wymagają adresu użytkownika w celu nawiązania z nim komunikacji.

Desktop Manager — moduł rozszerzający funkcje *Node Managera* systemu ANSA w zakresie sterowania urządzeniami, w które wyposażone są stacje multimedialne.

Translation Manager — moduł odpowiedzialny za rozwiązywanie niezgodności pomiędzy strumieniami danych poprzez wprowadzanie odpowiednich translatorów.

Wprowadzone nowe elementy architektury ANSA celowo jest odnieść do analizowanych w sekcji 2 pięciu składowych multimedialnych systemów komunikacyjnych. I tak składowej mediów dotyczą strumienie oraz urządzenia. Składowa komunikacyjna związana jest z QoS i zarządzaniem QoS, komponenty rozproszonego przetwarzania dotyczą *koordynacji* i własności odziedziczonych z bazowego systemu ANSA. Składowej aplikacyjnej dotyczą usługi *User Locatora*, *Desktop Managera* oraz *Translation Managera*.

Obecnie nie istnieje komercyjnie dostępna wersja systemu ANSA posiadająca proponowane rozszerzenia. Należy je zatem traktować jako wytyczne odnośnie do tworzenia aplikacji multimedialnych w istniejącej wersji oprogramowania. Większość z przedstawionych rozszerzeń da się bowiem zaimplementować bez trudu jako moduły aplikacji multimedialnej. Zasadnicza różnica polega tylko na mniejszej ogólności rozwiązania oraz zwiększonym nakładzie pracy w celu dostarczenia usług standardowo dostępnych w wersji multimedialnej systemu. Zostanie to zilustrowane w kolejnej sekcji na przykładzie implemetacji komunikacji głosowej w sieci Ethernet.

4. Architektura systemu komunikacji głosowej

Realizacja typowej aplikacji w środowisku ANSA składa się z następujących etapów:

1. zaprojektowanie modelu funkcjonalnego rzeczywistego systemu;
2. wyodrębnienie usług i obiektów obliczeniowych oraz specyfikacja interfejsów;
3. zaprojektowanie modelu inżynieryjnego i implementacja jego komponentów.

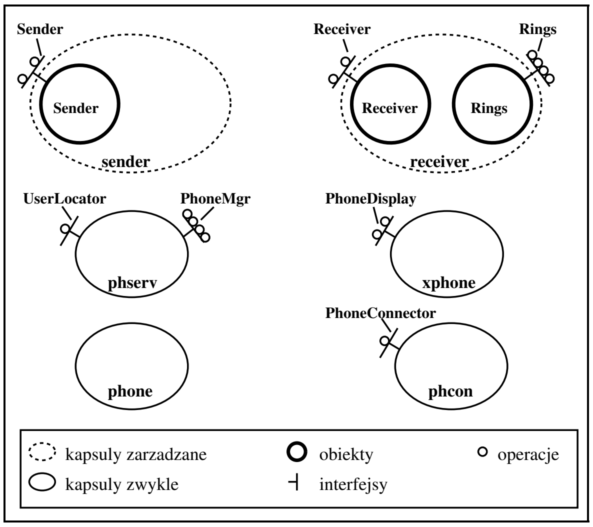
Ten modelowy schemat zastosowano w czasie konstrukcji systemu komunikacji głosowej w sieci Ethernet stacji roboczych SPARC, którego architektura jest tematem tej sekcji.

Założono, że projektowany system powinien umożliwiać komunikację głosową użytkowników lokalnej sieci komputerowej na zasadach analogicznych jak w sieci telefonicznej. Dlatego punkt wyjścia przeprowadzonych prac projektowych stanowiła analiza funkcjonalna prostej sieci telefonicznej składającej się z centrali oraz aparatów telefonicznych. W jej wyniku otrzymano model funkcjonalny rzeczywistego systemu telefonicznego.

Obiekty tego modelu zostały następnie odwzorowane na elementy architektury sprzętowej oraz na obiekty modelu obliczeniowego systemu ANSA. Otrzymany model obliczeniowy składał się z: serwera *phserv* spełniającego, w ramach wykonywanej usługi *PhoneMgr*, rolę centrali telefonicznej; serwerów *sender* nagrywających i wysyłających dźwięk (usługa *Sender*); serwerów *receiver* odbierających i odtwarzających dźwięk (usługa *Receiver*) oraz odtwarzających sygnały akustyczne (usługa *Rings*); klientów *phone* oraz *xphone* stanowiących interfejsy użytkownika. Model ten został następnie rozszerzony w sposób umożliwiający wykonywanie operacji niedostępnych w typowej sieci telefonicznej. W tym celu zaprojektowano dodatkowy obiekt obliczeniowy *phcon*, nowe usługi *PhoneConnector*, *UserLocator* i *PhoneDisplay* oraz dodano nowe operacje do usług zaprojektowanych wcześniej. Elementy modelu obliczeniowego zostały odwzorowane na elementy modelu inżynieryjnego: klienci i serwery na obiekty lub kapsuły, a usługi na interfejsy. Następnie wykonano implementację poszczególnych elementów, których struktura przedstawiona jest na rys. 1.

5. Implementacja systemu

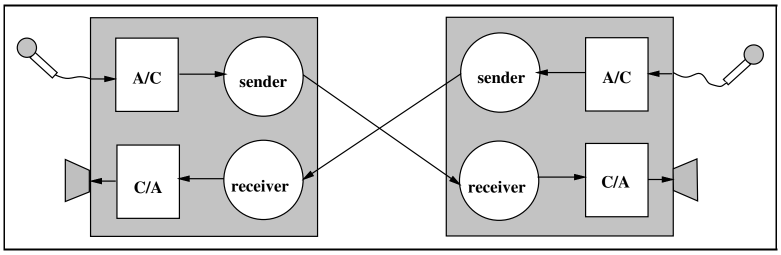
Platformą, na której zaimplementowano system, jest sieć stacji roboczych SPARC. Każda stacja wyposażona jest w urządzenie dźwiękowe złożone z przetwornika A/C i C/A, wbudowanego głośnika i podłączanego mikrofonu, które umożliwia przetwarzanie głosu do postaci cyfrowej i odwrotnie. Efektywne wykorzystanie tych urządzeń zapewniają kapsuły *sender* i *receiver*. Pierwsza z nich kieruje nagrywaniem oraz odczytuje z urządzenia kolejne porcje dźwięku i przesyła je za pomocą sieci komputerowej do uruchomionej na innej stacji kapsuły *reciever*. Kapsuła *receiver* odbiera porcje dźwięku i przesyła na wejście urządzenia dźwiękowego, które je odtwarza. Mikrofon, przetwornik A/C i kapsuła *sender* pełnią funkcje obwodu mikrofonu aparatu telefonicznego, a kapsuła *receiver*, przetwornik C/A i głośnik funkcje obwodu słuchawki. Podczas transmisji dwukierunkowej (rys. 2) między dwiema stacjami na każdej z nich musi być aktywna zarówno kapsuła *sender*, jak i *receiver*. Transmisja odbywa się w oparciu o mechanizm socketów i protokół UDP, który ze względu na małe narzuty jest bardziej odpowiedni do przesyłania głosu niż protokół TCP.



Rys. 1. Struktura elementów modelu inżynieryjnego systemu
Fig. 1. The structure of the system engineering model's items

Kapsuły *receiver* i *sender* odpowiadają za generację, transmisję i prezentację dźwięku. Dlatego ich interfejsy (*Receiver* i *Sender*) w omawianym w sekcji 2 rozszerzeniu systemu ANSA, mogłyby być zaimplementowane jako *urządzenia*. Rolę *strumienia* pełni datagramowe połączenie ustanawiane między kapsułami. Główny serwer aplikacji — kapsuła *phserv* — zamyka i otwiera połączenia odpowiednio zarządzając, za pośrednictwem kapsuł *receiver* i *sender*, urządzeniami dźwiękowymi i transmisją, a więc wykonuje funkcje *koordynacji*.

Sieć komputerowa stwarza możliwość poszerzenia zakresu usług oferowanych przez tradycyjną sieć telefoniczną, np. o operacje umożliwiające automatyczne wyszukiwanie użytkowników, rozgłaszanie komunikatów dźwiękowych, wyświetlanie identyfikatorów rozmówców (np. w formie fotografii) lub pozwalające na realizację połączeń telekonferencyjnych. W opisywanym systemie zaimplementowano trzy pierwsze usługi.

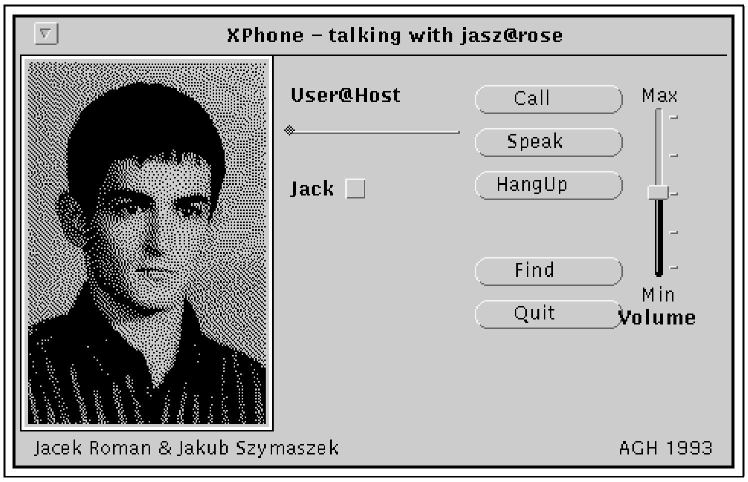


Rys. 2. Tor przepływu informacji akustycznej w czasie połączenia telefonicznego
Fig. 2. Ways of acoustic information flow during phone connection

Aby usunąć niedogodność posługiwania się nazwami komputerów zamiast identyfikatorami użytkowników, uzupełniono podstawowy model systemu o usługę lokalizacji użytkowników (*UserLocator*) w sieci komputerowej wykonywaną przez kapsułę *phserv*. Lokalizacja jest oparta na współpracy kapsuły zarządzającej z kapsułami *phcon* (*phone connector*). Na każdej stacji roboczej powinna zostać uruchomiona jedna taka kapsuła. Kapsuła *phserv* przesyła do każdej kapsuły *phcon* zapytanie, czy na jej stacji pracuje poszukiwany użytkownik. Jedynym zadaniem, jakie wykonują w systemie kapsuły *phcon*, jest udzielenie na to pytanie odpowiedzi.

Rozgłaszanie polega na transmisji dźwięku z jednej stacji roboczej do wielu innych. Implementacja tej usługi wymagała zaprojektowania kapsuły *sender* w taki sposób, aby mogła wysyłać nagrywany dźwięk do wielu kapsuł *receiver*. Ponadto konieczna była implementacja dodatkowych operacji głównego serwera, rozpoczynających i kończących rozgłaszanie.

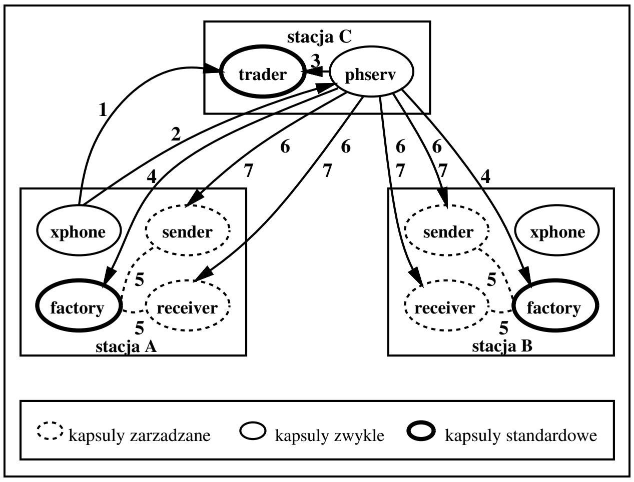
Kolejne rozszerzenie, jakie zrealizowano w środowisku X-Window, pozwala na wyświetlanie zdjęć rozmówców (rys. 3). Kapsuła *phserv* po zestawieniu połączenia przesyła odpowiednie fotografie do kapsuł *xphone*, które wyświetlają je na monitorach.



Rys. 3. Główne okno programu *xphone*
Fig. 3. The main window o the *xphone* program

Omówione usługi i komponenty systemu mogą być odwzorowane na usługi rozszerzonej wersji systemu ANSA w następujący sposób. Funkcję *User Locatora* wykonuje kapsuła *phserv* udostępniająca interfejs *UserLocator*. Sterowanie urządzeniami dźwiękowymi stacji roboczej, a więc funkcje *Desktop Managera*, wykonują kapsuły *receiver* i *sender*. Pozostałe dwie usługi rozszerzonego środowiska ANSA, tj *QoS Manager* i *Translation Manager*, ze względu na prostotę i homogeniczność aplikacji nie mają swoich odpowiedników.

Zasadę współdziałania poszczególnych komponentów przedstawiono na przykładzie operacji zestawiania połączenia między dwiema stacjami w sieci. Zestawianie połączenia składa się z dwóch etapów (rys. 4). W pierwszym przydzielane są wszystkie zasoby potrzebne do przeprowadzenia transmisji. W drugim jeden z użytkowników wyraża zgodę na przeprowadzenie rozmowy. Aby wywołać dowolną operację w kapsule *phserv*, użytkownik musi skorzystać z kapsuły *phone* lub *xphone* (które pełnią rolę interfejsów użytkownika) wybierając odpowiednią opcję lub komendę. Wówczas interfejs użytkownika komunikuje się ze standardową kapsułą ANSA — *traderem* i sprowadza z niej referencję usługi *PhoneMgr* (strzałka nr 1 na rys. 4), a następnie wywołuje wybraną operację tej usługi (strzałka nr 2).



Rys. 4. Schemat wykonania pierwszej fazy zestawiania połączenia między stacjami A i B
Fig. 4. The execution scheme of the connection establishment between A and B stations

Operacja zestawiania połączenia między stacjami A i B (rys. 4) rozpoczyna się od sprowadzenia z kapsuły *trader* referencji standardowej usługi *Factory* wykonywanej przez kapsuły *factory* znajdujące się na stacjach A i B (strzałka nr 3). Następnie w kapsułach *factory* wywoływana jest operacja (strzałki nr 4), która powoduje uruchomienie zarządzanych kapsuł *receiver* i *sender* na obu stacjach (linie przerywane nr 5). Każda z utworzonych kapsuł posiada standardowy interfejs *Capsule*, który wykorzystywany jest do utworzenia obiektów *Receiver*, *Rings* i *Sender* (strzałki nr 6). Następnie w kapsułach *receiver* i *sender* wywoływane są operacje (strzałki nr 7) inicjujące pracę urządzeń dźwiękowych, otwierające datagramowe połączenie transmisyjne i odtwarzające odpowiednie sygnały dźwiękowe (sygnały dzwonka na stacji B i oczekiwania na stacji A). Operację akceptacji połączenia inicjuje użytkownik stacji B. Powoduje to przerwanie odtwarzania sygnałów i rozpoczęcie transmisji głosu przez kapsuły *receiver* i *sender*.

5. Podsumowanie

Systemy multimedialne stawiają bardzo wysokie wymagania w zakresie szybkości transmisji danych, efektywności przetwarzania, synchronizacji i koordynacji wielu strumieni różnorodnej informacji. Większość istniejących środowisk programowania rozproszonego nie spełnia tych wymagań. W artykule przedstawiono krótko jedno z takich środowisk — system ANSA, a następnie omówiono koncepcję jego rozszerzenia o elementy ułatwiające konstruowane aplikacji multimedialnych. Jak jednak wykazano w ostatnich dwóch sekcjach, możliwie jest tworzenie takiego oprogramowania w oparciu o standardową wersję systemu ANSA. Okupione jest to jednak dużym nakładem pracy, ponieważ należy zaimplementować usługi i zrealizować komponenty standardowo dostępne w rozszerzonej wersji środowiska.

LITERATURA

[1] ANSA: An engineer's introduction to the architecture. APM Ltd, Cambridge 1989.

[2] ANSAware 4.0 application programer's manual. APM Ltd, Cambridge 1989.

[3] ANSAware 4.0 system programer's manual. APM Ltd, Cambridge 1989.

[4] Bijałd M., Gwóźdź M., Zieliński K.: Rozproszone przetwarzanie w systemach otwartych — model ANSA. Informatyka, 1993, Nr 3.

[5] Bijałd M., Gwóźdź M., Zieliński K.: ANSA. Rozdział w książce: Środowiska programowania rozproszonego w sieciach komputerowych. Księgarnia Akademicka, Kraków 1994, s. 116-135.

[6] Hopper A.: Pandora — an experimental system for multimedia applications. ACM Operating Systems Review, 1990, No. 2.

[7] Nicolaou C. A.: A distributed architecture for multimedia communication systems. Technical Report No. 220, Universisty of Cambridge Computer Laboratory, Cambridge 1991.

[8] Vin H.M., Chen M., Barzilai T.: Collaboration management in DiCE. The Computer Journal, 1993, No. 2.

[9] Zellweger P.T., Terry D.B., Swinehart D.C.: An overview of the Etherphone system and its applications. Materiały konferencji 2nd IEEE Conference on Computer Workstations, str. 160-168, 1988.

Recenzent: Dr hab. inż. Tadeusz Czachórski

Wpłynęło do Redakcji 21 listopada 1994 r.

Abstract

Contemporary environments for distributed programming, such as ANSA (Advanced Networked Systems Architecture), often do not comply with requirements of multimedia systems. In this paper, concepts of ANSA programming environment's extensions, proposed and described in details in a thesis [7], enabling easy multimedia applications programming, are presented. These extensions include new elements of programming languages and new services of the environment. Extensions of programming languages facilitate design and description of components (such as media streams, devices) and functions (such as coordination of many data streams in various forms, e.g. voice or motion video, with different quality of service demands) specific for multimedia systems. The new services enables management of devices and streams, translation of media streams coded in different formats, users' location, etc.

Although none of these extensions have been implemented, elements of the described concepts may be used to build multimedia applications based on a currently version of ANSA environment, as shown in the last two sections of this paper on the example of voice communication system. The implemented system enables phone calls, voice broadcasting, user location and displaying of talking users' pictures. It is composed of: a main server *phserv* playing a role of a telephone exchange; *receiver* and *sender* servers that manage voice playing, transmission and recording (Fig. 2); a user interface *phone*; a graphical user interface *xphone* (Fig. 3); and *phcon* server searching for users in cooperation with *phserv* server. The structure of the application's components is shown in Fig. 1. The execution scheme of connection establishment between two workstations, presented in Fig. 4, illustrates cooperation of the system's and ANSA environment's items.