DIPE: A Distributed Environment for Medical Image Processing

Marios ZIKOS[1,2], Eleni KALDOUDI[1], Stelios C. ORPHANOUDAKIS[1,2]
[1]*Institute of Computer Science, FORTH, P.O. Box 1385, 711 10 Heraklion, Greece*
[2]*Department of Computer Science, University of Crete, Heraklion, Greece*
E-mail: {zikos, kaldoudi, orphanou}@ics.forth.gr

Abstract. DIPE is a distributed environment that provides image processing services over integrated teleradiology services networks. DIPE integrates existing and new image processing software and employs sophisticated execution scheduling mechanisms for the efficient management of computational resources within a distributed environment. It can also be extended to provide various added-value services, such as management and retrieval of image processing software modules, as well as advanced charging procedures based on quality of service. DIPE can be viewed as the natural evolution of the legacy field of medical image processing towards a service over the emergent health care telematics networks.

1. Introduction

In recent years, advances in information technology and telecommunications have acted as catalysts for significant developments in the sector of health care. These technological advances have had a particularly strong impact in the field of medical imaging, where film radiographic techniques are gradually being replaced by digital imaging techniques, and this has provided an impetus to the development of integrated hospital information systems and integrated teleradiology services networks which support the digital transmission, storage, retrieval, analysis, and interpretation of distributed multimedia patient records [1]. One of the many added-value services that can be provided over an integrated teleradiology services network is access to high-performance computing facilities in order to execute computationally intensive image analysis and visualisation tasks [2].

In general, currently available products in the field of image processing (IP) meet only specific needs of different end user groups. They either aim to provide a comprehensive pool of ready to use software within a user-friendly and application specific interface for those users that use IP software, or aim for the specialised IP researcher and developer, offering programmer’s libraries and visual language tools. However, we currently lack the common framework that will integrate all prior efforts and developments in the field and at the same time provide added-value features that support and in essence realise what we call a ‘service’. In the case of image processing, these features include: computational resource management and intelligent execution scheduling; intelligent and customisable mechanisms for the description, management, and retrieval of image processing software modules; mechanisms for the “plug-and-play” integration of already existing heterogeneous software modules; easy access and user transparency in terms of software, hardware, and network technologies; sophisticated charging mechanisms based on quality of service; and, methods for the integration with other services available within an integrated health telematics network.

In this paper we present the architecture of DIPE, a novel distributed environment for image processing services. DIPE is based on a distributed, autonomous, co-operating agent architecture [3]. It is designed so that it is modular, scaleable and extensible, and it can be readily implemented on different hardware and software platforms, and over heterogeneous networks. DIPE consists of a functional core which supports the persistent distributed execution of IP algorithms, and can be extended to support other added-value services such as macros, resource management, algorithm retrieval, charging, etc. Here we describe the functional core of the system and discuss the mechanisms and notions employed to allow integration of third party IP algorithms and the development of new IP software. Finally, we describe the functional extensions of the core that support macro execution and resource management. DIPE has been developed to support distributed medical imaging processing, an added-value teleradiology service within the integrated regional health telematics network, currently under development by the Institute of Computer Science (ICS), Foundation for Research and Technology - Hellas (FORTH), on the island of Crete [4].

2. Architecture and Implementation

The core of the system consists of several communicating components: user applications, execution agents, pools of IP algorithms, and management agents.

The management agent is the central element. Its main purpose is to realise the network of individual modules (applications and execution agents) and initialise the communication among them. However, the main body of messages is communicated directly among the individual modules. The local cluster can be further expanded through a network of management agents, within the same or even different organisations. Thus, the management agent ensures the scaleability of the environment, a basic requirement of an integrated teleradiology services network [1]. Additionally, the management agent authenticates users and provides unique image ids by using standard digital signature technology.

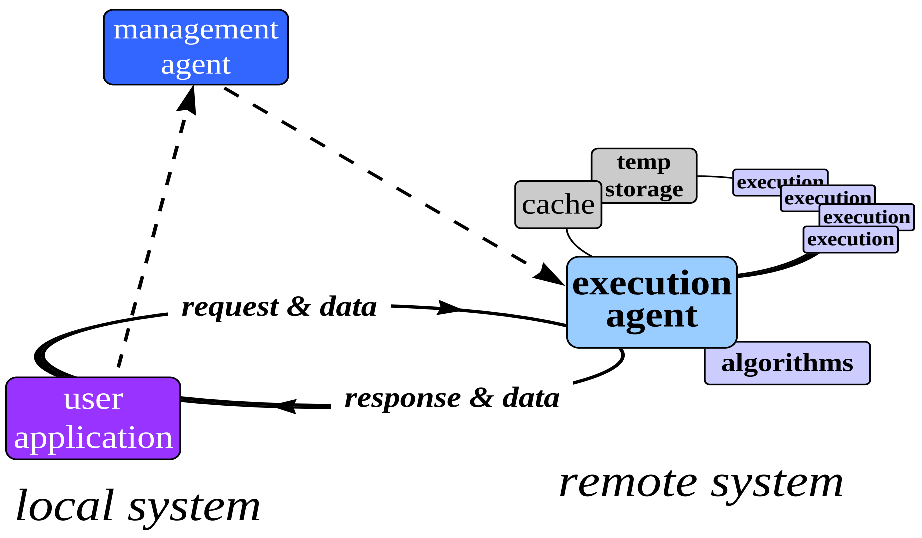
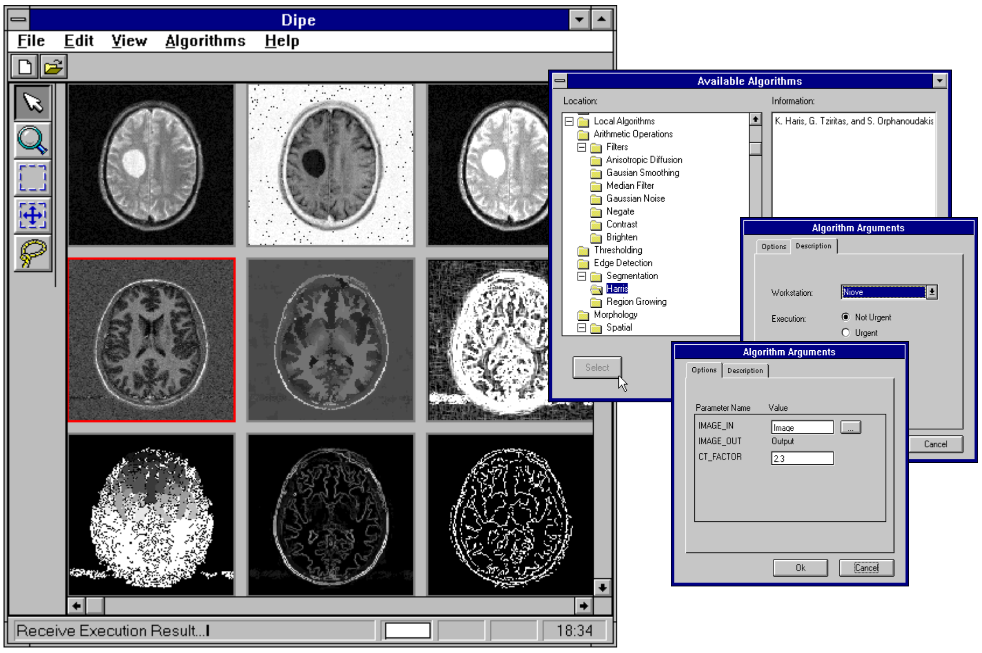


Figure 1: Communication within a DIPE cluster

The execution agent is responsible for the execution of a specific algorithm. It receives requests for execution through the management agent and creates a communication link with the requesting application in order to receive further information and input data required for the execution (Figure 1). After this point, this agent can proceed autonomously to the execution of the algorithm. It stores input data into a local cache area and executes the requested algorithm. Output generated through the execution of the algorithm is sent back to the agent. The execution agent is responsible to forward this output to the requesting application. In case there is a network failure or the requesting application is not running any longer, the agent keeps the results of the execution in temporary storage for delivery upon request. This ensures persistent algorithm execution and enhances the robustness of the system.

The user application is the front end of the system and consists primarily of a customisable graphical user interface. A virtual temporary storage management module ensures that the application can handle synchronously a considerable number of large data sets. An important feature of the user application is that it incorporates certain image processing algorithms that require real-time response, and thus it is not sensible to redirect their execution to an agent or over the network. These include routines necessary for image visualisation (e.g., zoom, focus, resize, contrast adjustment, etc.), as well as certain algorithms for local, real-time image processing. Finally, the graphical user interface provides toolkits that support the various functionalities of the environment (algorithm insertion, monitoring of the system’s status, resource management, macro composition and execution, etc.). A typical screen of the application is shown in Figure 2.

The basic requirement that DIPE is readily implemented on various operating systems and over heterogeneous networks poses certain implementation constraints. Thus, inter-process communication is based on the TCP/IP network protocol, while operating system transparency is ensured by using ACE, an object-oriented network programming toolkit for developing communication software [5]. DIPE is now implemented on UNIX and Windows NT/95 workstations.



K. Haris, G. Tziritas, and S. Orphanoudakis

Workstation: Niove

Execution: [✓ Not Urgent] Urgent

Parameter Name	Value
IMAGE_IN	Image
IMAGE_OUT	Output
CT_FACTOR	2.3

Receive Execution Result...| 18:34

Figure 2: A typical screen of DIPE

3. The Algorithm Repository

The functional core of DIPE is the set of available image processing algorithms, private or public, local or network wide. An important feature of DIPE is that it allows easy integration of third party algorithms, i.e. software modules where only an executable is available and the only information known is the command line syntax, as well as the input and output data formats. The integration is achieved through the algorithm wrapper, a single generic process. The wrapper converts input data from the application format to the format that a specific IP algorithm requires, executes the algorithm and finally converts the output data of the algorithm to the format of the user application. While the algorithm is being executed, the wrapper is responsible to handle requests from the user application. Such requests include the termination or pause of the execution, or the resumption of a previously paused execution. Additionally, DIPE provides a library of ready-to-use routines for the development of new IP algorithms, which consists of basic routines related to the starting and ending phases of the algorithm, as well as of routines that support a more sophisticated mode of user-algorithm communication during execution.

In routine medical image processing, a common situation involves processing images using the same set of algorithms often with a standard set of parameter values. DIPE provides the mechanisms to simplify the complicated process of executing individual algorithms sequentially, by grouping them together and thus creating a macro-algorithm (macro). In general, the DIPE macro is a set of individual algorithms that may be performed independently on the same or different data sets, or may be performed sequentially. There is no constraint on the complexity of algorithm combinations and the inter-relationships of their input and output data. The execution of a macro is the responsibility of a special macro agent. The macro agent acts as a mediator for macro executions. It consists of three main functional parts: the interface with the application, the interface with the rest of the system (management and execution agents), and the module which is responsible for the management of the macro execution. The macro agent models macros as a directed acyclic graph, thus enabling macro decomposition and individual scheduling of its components.

4. Resource Management

Quality of service in DIPE is guaranteed by a sophisticated resource management and execution scheduling mechanism. The scheduling of a requested algorithm execution to the most appropriate processing element (PE) is a distributed decision making process based on the market metaphor, and is realised through the co-operation of the execution agents [6, 3]. Upon request for an algorithm execution, the management agent initialises an ‘auction’. The request is forwarded to the appropriate ‘bidders’, that is those execution agents that are able to perform the request. Each execution agent evaluates the request by taking into consideration the load of the local PE, the possible existence of the required input data in its local cache vs. the cost for transferring the data through the network, and the execution characteristics of the particular algorithm. Then, each execution agent makes a bid to the management agent by returning the estimated ‘cost’ of the execution. The management agent evaluates all the bids it receives and assigns the execution to a particular execution agent.

It is important to note that the execution characteristics of each algorithm are drawn from its execution profile, which includes information on size of input/output data, PE memory needed at runtime (relative to input data) and time needed for execution (normalised to input data and PE). A good approximation about the memory requirements and the execution time of an algorithm is derived from a statistical analysis based on previous execution profiles of the algorithm.

5. Discussion

DIPE has been designed and developed to offer image processing services over integrated health care services networks, and to act as an integration platform for diverse image processing software. It exhibits a modular, extensible and scaleable architecture that ensures system robustness and execution persistence. A sophisticated resource management and execution scheduling mechanism allows the medical expert to take full advantage of geographically distributed computational resources. Future research will address the development of intelligent and customisable mechanisms for the description, management, and retrieval of image processing software modules, as well as charging mechanisms based on quality of service.

DIPE is currently being extended through its functional integration with other medical information systems that have been developed in our laboratory. Important examples include CoMed [7], a desktop conferencing application which allows interactive real-time co-operation among several medical experts, as well as TelePACS [4], an information system for medical image management and communication. DIPE is one of the diverse telematics applications incorporated in the regional health telematics network, which is currently being developed by ICS-FORTH on the island of Crete.

6. References

[1] S.C. Oprhanoudakis, E. Kaldoudi, and M. Tsiknakis, “Technological Advances in Teleradiology”, Eur. J. Radiology, vol. 22, 205-217, 1996.

[2] S.C. Orphanoudakis, “Supercomputing in Medical Imaging” IEEE Eng Med Biol, vol. 7, 16-20, 1988.

[3] P. Maes, “Modelling Adaptive Autonomous Agents”, Artificial Life Journal, ed. C. Langton, vol. 1, nos. 1&2, MIT Press, 1994.

[4] S.C. Orphanoudakis, M. Tsiknakis, C. Chronaki, S. Kostomanolakis, M. Zikos, and Y. Tsamardinos, “Development of an Integrated Image Management and Communication System on Crete”. In: Lemke HU, Inamura K, Jaffe CC, Vanier MW, eds. Proc. of CAR’95, Berlin, p. 481-487, 1995.

[5] D.C. Schmidt, “The ADAPTIVE Communication Environment: An Object-Oriented Network Programming Toolkit for Developing Communication Software”, 12th Sun User Group Conference, San Francisco, California, June 14-17, 1993.

[6] D.F. Ferguson, Y. Yemini, C. Nikolaou, “Microeconomic Algorithms for Load Balancing in Distributed Computer Systems.”, . In Proceedings of International Conference on Distributed Systems (ICDCS 88). San Jose, California: IEEE Press, 1988.

[7] M. Zikos, C. Stephanidis, and S.C. Orphanoudakis, “CoMed: Cooperation in Medicine”, Proceedings of EuroPACS’96, pp. 88-92, Heraklion, Greece, October 3-5, 1996.