Medical Informatics Europe 2002
G Surján et al. (eds)
IOS Press 2002

Design of Acquisition Devices Management Subsystem for IEEE 1073 Compliant Software Agents

V. A. NAGIN, I. V. POTAPOV, S. V. SELISHCHEV

Moscow State Institute of Electronic Technology, Moscow, 103498, K-498, Russia.

Abstract. The paper addresses the issue of device management system design for software agents compliant with IEEE 1073 device communication standard. Based on middleware architecture the device control layer represents a universal versatile object-oriented application-programming interface. The approach presented in the paper allows to implement plug-and-play integration and interoperability of medical acquisition devices within the medical device system be means of common middleware services. Adherence to Medical Data Information Base nomenclature, component part of IEEE 1073 communication standard, adds necessary consistency to presented component-based infrastructure.

1. Introduction

One of the most complicated problems in the healthcare apart from information system integration is the interoperability of patient connected devices. A number of medical systems such as intensive care units, specialized diagnostic and monitoring software, home-based medical systems are affected by acquisition device interoperability issues. The core of the so-called plug-and-play interoperability problem is to provide standard all-sufficient communication interface between a medical device and any device or a system providing all necessary features for device operation. IEEE 1073 family of standards for medical device communications, known as Medical Information Bus (MIB), addresses the medical device interoperability problem [1]. The aim of the standard is to bring a wide range of medical devices under its purview providing transparent plug-and-play interoperability and ease of reconfiguration. Communication process is seen as a logical interface between two systems: a manager (typically a host represented by bedside communication controller) and an agent (Fig. 1).

An agent application process is an abstract name for a medical device system, which provides data in a manager/agent communicating system. Communication protocol is built upon the standard ISO/OSI layers and established by means of Association Control Service Element (ACSE)[2] providing logical connection between the communicating parties and Common Medical Device Information Service Element (CMDISE)[3] enabling access to the internal object infrastructure of Medical Data Information Base (MDIB). MDIB contains the structure of the managed medical objects as defined in the Domain Information Model of IEEE 1073. These managed medical objects represent the "real world instances" of the medical application domain, which includes Medical Device System (MDS), Virtual Medical Device (VMD), Channel and Real Time Sample Arrays (RTSA). IEEE 1073 and its accompanying standards tend, however, to cover all topics of medical device engineering what, in turn, makes implementation of agent systems very complicated.

Moreover, the standards define only high-level interfaces for MDIB access leaving the burden of implementation of object management to developers.

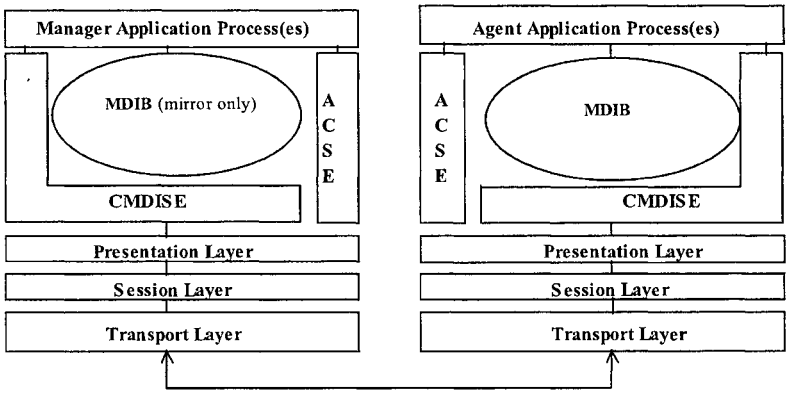

Fig. 1: Communicating Systems - Manager/ Agent Framework

The aim of this paper is to investigate currently available middleware services, substantiate the necessity for device management subsystem, draw basic design concept for medical device management layer and present the implementation of the corresponding object-oriented infrastructure.

2. Design concepts

Middleware technologies have proved to be a universal and flexible solution for the variety of distributed software systems in different application spheres and in particular in medical systems integration [4]. A key factor of its popularity is the ability of middleware to solve specific interoperability problems in the distributed and heterogeneous environments. Middleware services, dedicated software modules designed according to the common technology specifications, form the protocol stack of information layers over the operating system transport layer below specific applications [5].

As noted in previous paragraph, implementation of an agent implies the development of required by IEEE 1073 communication and data processing layers for each device in a medical device system (Fig. 2). A medical device system may comprise one or more devices and "in a real world" may correspond to any acquisition or control system. The use of proper middleware component infrastructure which, in fact, appears to be an object-oriented Application Programming Interface (API) allows to simplify significantly the implementation of data agents by means of utilization of the common hardware abstraction layer for the number of services be it MIB communication controller or medical stand-alone application.

The basic functionality required from the dedicated device management layer by acquisition systems includes:

1) The unified API for enumeration, access and management of available connectable devices;
2) Support of plug-in-play integration of the new acquisition device "on the fly" in the hosting system;
3) An object-oriented data model;
4) Universal data transport and alert signaling protocol.

A comparative study of the most widespread software component client/server platforms including Component Object Model (COM+)[6] and Common Object Request Broker Architecture (CORBA), two very popular middleware technologies, has been carried out. Fulfillment of several general requirements was investigated during the study. General criteria of evaluation of middleware platform for successful implementation of device management subsystem include: reliability, high performance for real-time applications, high availability, and support of component infrastructure. Despite less broad acceptance by industry, COM+ showed some advantages over CORBA owing to provision of its own component model for manageable software modules and good real-time qualities.

Basic concepts of the selected component platform led to the following design principles for the device management system architecture:

1) Mapping of Medical Data Information Base object-oriented concept onto COM+ interfaces.
2) Conformance to basic requirements of Healthcare Information Systems Architecture (HISA)[7].
3) Implementation of unified data management;
4) Open interface implementation for seamless integration of third party components into a the system.

The conceptual diagram of the device management layer architecture is presented on the Fig. 2.

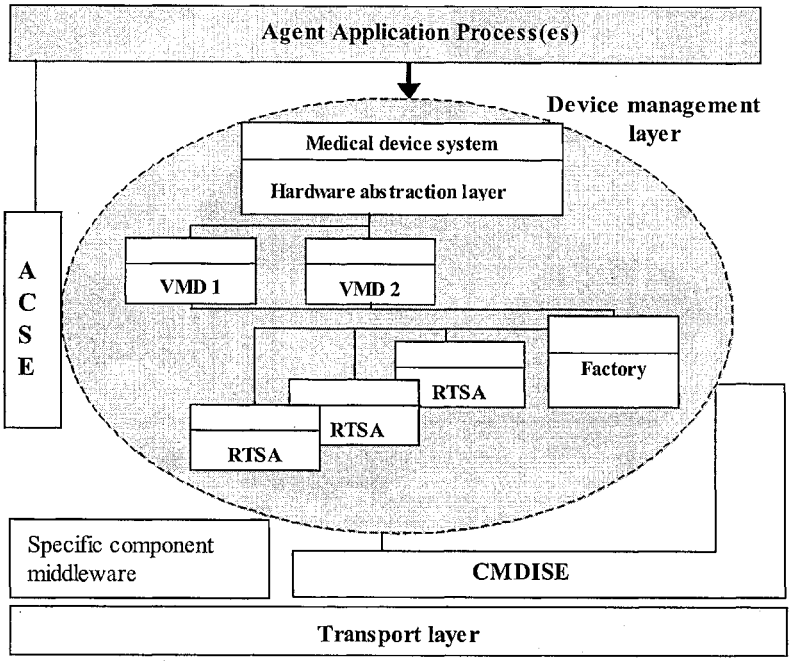

Fig. 2: Conceptual architecture of COM-based software agents.

Mapping of MDIB object models onto COM+ interface specification has resulted in several fundamental types of components representing corresponding MDIB abstractions: medical device system, virtual medical device (VMD), real-time sample arrays (RTSA), alert objects and sample array factory. An application process built on top of middleware infrastructure utilizes the object-oriented hardware abstraction layer for interaction with

attached devices, which provides all necessary functionality for hosting applications. Such abstraction layer is conventionally called medical device system. Real-time sample array objects are used to transfer any real-time data, alert objects - for describing different events appearing in the system. Sample arrays and alert objects are created and managed by sample factory, which serves as a garbage collector manager.

Virtual medical devices might be implemented as a stand-alone COM+ and provide interfaces offering following functionality:

1) Transport of acquired real-time sample arrays and alert objects;
2) Universal identification of the device;
3) Basic setup and control functions;
4) Alert mechanism.

Two options exist to access the device management layer functionality from the network: by means of standard remote access services offered by middleware or by the general implementation of the common medical device information service element (CMDISE). It provides basic services for managed objects, including the GET, SET, CREATE, DELETE, ACTION, and EVENT REPORT service functions.

3. Implementation

A laboratory ECG diagnostic system IntelliCard was developed on top COM+ infrastructure of device management subsystem including several components, which can be used as building blocks for any real-time ECG system [5]. The system conforms to the three-tier software architecture (data, business, presentation tier) also known as Windows DNA – Distributed interNet Applications Architecture (Fig. 2). Device management layer constitutes the data tier. Business tier components responsible for data handling and processing are implemented as pure COM components with support of automation interfaces; presentation tier contains fully functional ActiveX components supporting common manipulation interfaces [8,9]. Microsoft VC6.0 with the accompanying COM development libraries was used to develop the host system and entire object infrastructure.

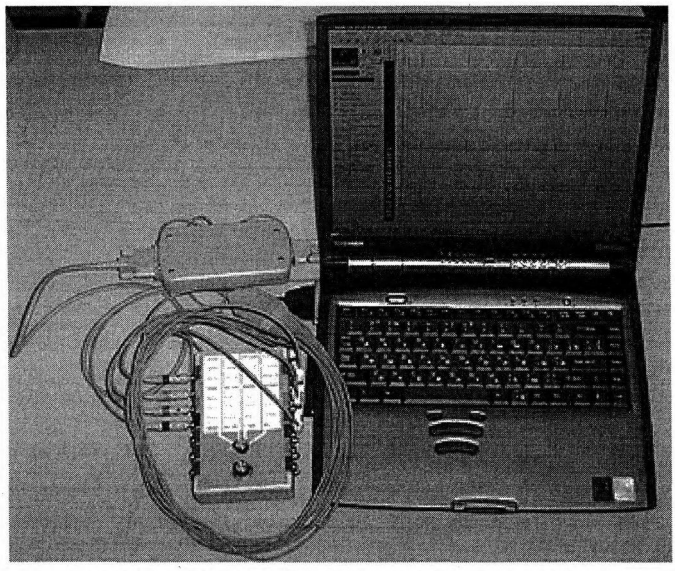

Fig. 3: IntelliCard ECG acquisition and processing system.

The system consists of ECG amplifier, interface PCMCIA board, notebook computer and optional ECG signal generator.

The software provides measurement and analysis features of:

1) Standard 12 channel ECG;
2) 6 channel ECG;
3) 3 channel ECG.

The ECG is measured with 500Hz sampling rate and 22bit Analog-to-Digital conversion [11] what guaranties very high recording accuracy for further analysis. The interface part of the ECG acquisition device is implemented as PCMCIA card; therefore measurements are possible both with desktop PC systems and mobile computers.

Virtual device objects developed for such system include ECG acquisition device component as well as blood pressure monitor and pulse oximeter modules.

4. Discussion

The presented paper concentrates attention on the internal design of the data acquisition agents, or medical device systems, in terms of MDIB. The concept of the device management layer presented here is not limited to the computer electrocardiography domain. It is worth noting that implementation of dedicated device services makes sense for the applications which are intended to support a number of similar acquisition devices designed for same purposes by different manufactures and thus having manifold protocols and physical interfaces.

5. Conclusion

The COM+ technology appears to be a very effective tool for building multifunctional distributed medical applications component-wise. Costs of supporting COM architecture in terms of the CPU utilization time and the additional source code overhead are relatively low. Provided that a basic system infrastructure is implemented and tested, development and integration of new components can be done relatively easy.

References:

[1] Standard for Medical Device Communications. Overview and Framework, IEEE 1073 –1996 (The Institute of Electrical and Electronics Engineers, 1996).

[2] ISO 8649, Information technology - Open Systems Interconnection – Service definition for the Association Control Service Elements

[3] ISO/IEC 8650, Information technology - Open Systems Interconnection –Specification of Protocols for Common Application Service Elements.

[4] Katehakis D. G., Tsiknakis M., Armaganidis A., Orphanoudakis S. C., "Functional and Control Integration of an ICU, LIS and PACS Information System", Proceedings of MIE'97, IOS Press, Porto Carras, Greece, May 25-29, 1997, pp. 15-19.

[5] Information Technology—Open Systems Interconnection—Basic Reference Model: The Basic Model, ISO/IEC 7498-1 (Geneva: International Organization for Standardization, 1994).

[6] Short Strategic Study: Enabling Technologies – CORBA and COM/DCOM, Final Report, TC251 Secretariat, Brussels, Belgium, CEN/TC251/N98-108,1998.

[7] European Prestandard ENV 12967-1:1998 Medical Informatics – Healthcare Information System Architecture (HISA) – Part 1: Healthcare Middleware Layer. CEN TC/251, 1998.

[8] Nagin V.A., Potapov I.V., Selishchev S.V., Sharf V. A software package of acquisition and mathematical processing of ECG data on the basis of COM architecture. Meditsinskaya Technika, vol 1, Moscow–2001. –P.3-7. English translation is available in Biomedical Engineering published by Kluwer Publishers

[9] V.A. Nagin, I.V. Potapov, S. V. Selishchev . A Distributed Component-Oriented Architecture for Real-Time ECG Data Acquisition Systems. CD-ROM Proceeding of 23rd IEEE International Conference on Engineering in Medicine and Biology.- Istanbul, October 25-28, 2001.

[10] V.A. Nagin, I.V. Potapov, S. V. Selishchev . A Universal Object-Oriented Medical Database System. CD-ROM Proceeding of 23rd IEEE International Conference on Engineering in Medicine and Biology.- Istanbul, October 25-28, 2001.

[11] Kurekov S. F., Prilutski D. A., Selishchev S. V. Sigma-Delta Analoque-to-Digital converters for Biomedical Data Acquisition Systems // Proceeding of 4th European conference on engineering and medicine.- Warsaw, May 25-28, 1997.-P.163-164.