Virtual Instrumentation

Željko Obrenovic, Dušan Starcevic, Emil Jovanov

1. Introduction

Virtual instrumentation is an interdisciplinary field that merges sensing, hardware and software technologies in order to create flexible and sophisticated instruments for control and monitoring applications. There are several definitions of a virtual instrument available in the open literature. Santori defines a virtual instrument as "an instrument whose general function and capabilities are determined in software" [Santori91]. Goldberg describes that "a virtual instrument is composed of some specialized subunits, some general-purpose computers, some software, and a little know-how" [Goldberg00]. Although informal, these definition capture the basic idea of virtual instrumentation and virtual concepts in general - provided with sufficient resources, "any computer can simulate any other if we simply load it with software simulating the other computer" [Denning01]. This universality introduces one of the basic properties of a virtual instrument – its ability to change form through software, enabling a user to modify its function at will to suit a wide range of applications. The concept of *virtual instrumentation* was born in late 1970s, when microprocessor technology enabled a machine's function to be more easily changed by changing its software [Santori91]. The flexibility is possible as the capabilities of a virtual instrument depend very little on dedicated hardware - commonly, only application-specific signal conditioning module and the analog-to-digital converter used as interface to the external world. Therefore, simple use of computers or specialized onboard processors in instrument control and data acquisition cannot be defined as virtual instrumentation.

Increasing number of biomedical applications use virtual instrumentation to improve insights into the underlying nature of complex phenomena and reduce costs of medical equipment and procedures [Loob00]. Although many of the general virtual instrumentation concepts may be directly used in biomedical measurements, the measurements in the medical field are peculiar as "they deal with a terribly complex object— *the patient* —and are performed and managed by another terribly complex instrument — *the physician*" [Parvis02].

In this chapter we describe basic concepts of virtual instrumentation, as well as biomedical applications of virtual instrumentation. In the second section we give a brief history of virtual instrumentation. The architecture of a virtual instrument and contemporary development tools are described in the third section. In the fourth section we describe the organization of the distributed virtual instrumentation. Finally, we present some biomedical applications of virtual instrumentation.

2. A Brief History of Virtual Instrumentation

A history of virtual instrumentation is characterized by continuous increase of flexibility and scalability of measurement equipment. Starting from first manual-controlled vendor-defined electrical instruments, the instrumentation field has made a great progress toward contemporary computer-controlled, user-defined, sophisticated measuring equipment. Instrumentation had the following phases:

- Analog measurement devices,
- Data Acquisition and Processing devices,
- Digital Processing based on general purpose computing platform, and
- Distributed Virtual Instrumentation.

The first phase is represented by early "pure" analog measurement devices, such as oscilloscopes or EEG recording systems. They were completely closed dedicated systems, which included power suppliers, sensors, translators and displays [Geddes89]. They required manual settings, presenting results on various counters, gauges, CRT displays, or on the paper. Further use of data was not part of the instrument package, and an operator had to physically copy data to a paper notebook or a data sheet. Performing complex or automated test procedures was rather complicated or impossible, as everything had to be set manually.

Second phase started in 1950s, as a result of demands from the industrial control field. Instruments incorporated rudiment control systems, with relays, rate detectors, and integrators. That led to creation of proportional-integral-derivative (PID) control systems, which allowed greater flexibility of test procedures and automation of some phases of measuring process [Goldberg00]. Instruments started to digitalize measured signals, allowing digital processing of data, and introducing more complex control or analytical decisions. However, real-time digital processing requirements were too high for any but an onboard special-purpose computer or digital signal processor (DSP). The instruments still were standalone vendor defined boxes.

In the third phase, measuring instruments became computer based. begun to include interfaces that enabled communication between the instrument and the computer. This relationship started with the general-purpose interface bus (GPIB) originated in 1960's by Hewlett-Packard (HP), then called HPIB, for purpose of instrument control by HP computers. Initially, computers were primarily used as off-line instruments. They were further processing the data after first recording the measurements on disk or type [Nebeker02].

As the speed and capabilities of general-purpose computers advanced exponentially general-purpose computers became fast enough for complex real-time measurements. It soon became possible to adapt standard, by now high-speed computers, to the online applications required in real-time measurement and control. New general-purpose computers from most manufactures incorporated all the hardware and much of the general software required by the instruments for their specific purposes. The main advantages of standard personal computers are low price driven by the large market, availability, and standardization.

Although computers' performance soon became high enough, computers were still not easy to use for experimentalists. Nearly all of the early instrument control programs were written in BASIC, because it had been the dominant language used with dedicated instrument controllers. It required engineers and other users to become programmers before becoming instrument users, so it was hard for them to exploit potential that computerized instrumentation could bring. Therefore, an important milestone in the history of virtual instrumentation happened in 1986, when National Instruments introduced LabVIEW 1.0 on a PC platform [Santori91]. LabVIEW introduced graphical user interfaces and visual programming into computerized instrumentation, joining simplicity of a user interface operation with increased capabilities of computers. Today, the PC is the platform on which most measurements are made, and the graphical user interface has made measurements user-friendlier.

As a result, virtual instrumentation made possible decrease in price of an instrument. As the virtual instrument depends very little on dedicated hardware, a customer could now use his own computer, while an instrument manufactures could supply only what the user could not get in the general market.

The fourth phase became feasible with the development of local and global networks of general purpose computers. Since most instruments were already computerized, advances in telecommunications and network technologies made possible physical distribution of virtual instrument components into telemedical systems to provide medical information and services at a distance. Possible infrastructure for distributed virtual instrumentation includes the Internet, private networks and cellular networks, where the interface between the components can be balanced for price and performance [Goldberg00].

3. Virtual Instrument Architecture

A virtual instrument is composed of the following blocks:

- Sensor Module,
- Sensor Interface,
- Medical Information Systems Interface,
- Processing Module,
- Database Interface, and
- User Interface.

Figure 1 shows the general architecture of a virtual instrument.

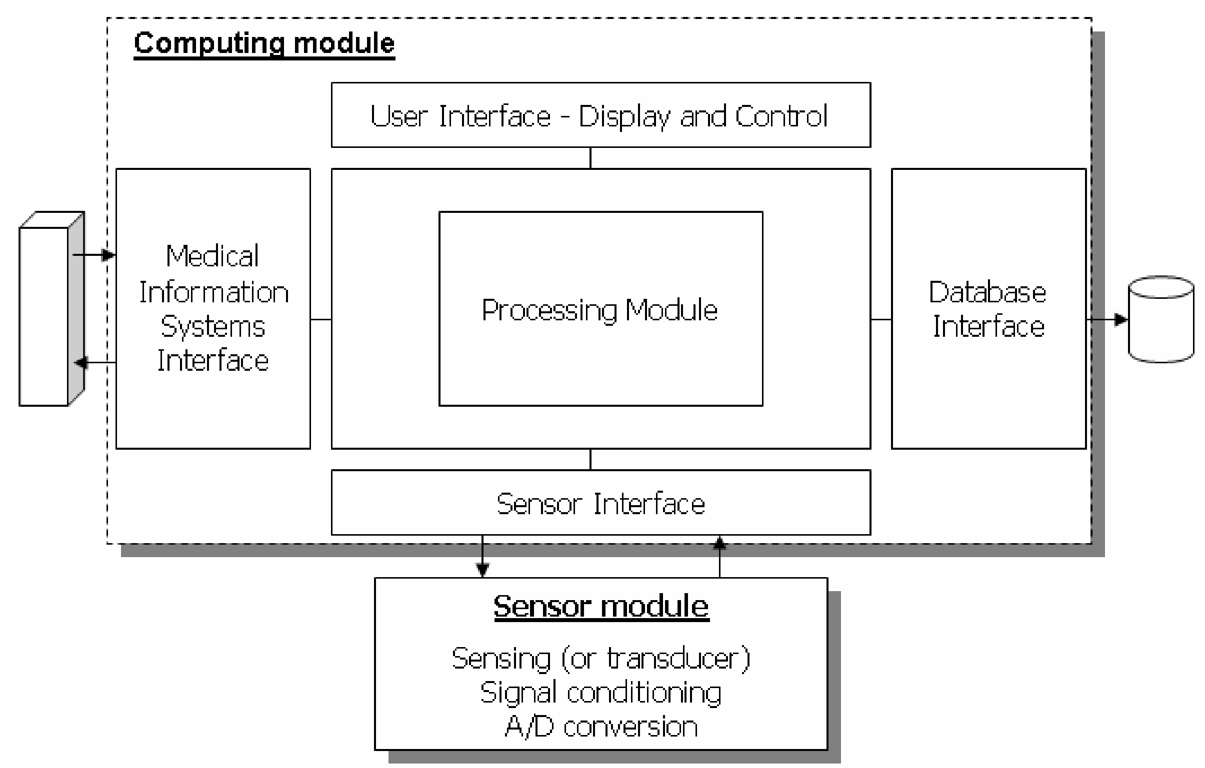

Figure 1: *Architecture of a virtual instrument.*

The sensor module detects physical signal and transforms it into electrical form, conditions the signal, and transforms it into a digital form for further manipulation. Through a sensor interface, the sensor module communicates with a computer. Once the data are in a digital form on a computer, they can be processed, mixed, compared, and otherwise manipulated, or stored in a database. Then, the data may be displayed, or converted back to analog form for further process control. Biomedical virtual instruments are often integrated with some other medical information systems such as hospital information systems. In this way the configuration settings and the data measured may be stored and associated with patient records.

In following sections we describe in more details each of the virtual instruments modules.

3.1. Sensor module

The sensor module performs signal conditioning and transforms it into a digital form for further manipulation. Once the data are in a digital form on a computer, they can be displayed, processed, mixed, compared, stored in a database, or converted back to analog form for further process control. The database can also store configuration settings and signal records.

The sensor module interfaces a virtual instrument to the external, mostly analog world transforming measured signals into computer readable form. Table # summarizes some of the often used human physiological signals [Charles99].

Table #: *Human physiological signals.*

Group	Physiological signals
Electrical signals (require only amplification)	Electromyograph (EMG)
	Electrocardiograph (ECG)
	Electroencephalograph (EEG)
	Electrooculograph (EOG).
Non-electrical signals (require a transducer to change the information to an electrical signal)	Skin conductivity (Galvanic Skin Response - GSR)
	Respiratory rate
	Blood pressure
	Peripheral body temperature

A sensor module principally consists of three main parts:

- the sensor,
- the signal conditioning part, and
- the A/D converter.

The sensor detects physical signals from the environment. If the parameter being measured is not electrical, the sensor must include a transducer to convert the information to an electrical signal, for example, when measuring blood pressure. According to their position, biomedical sensors can be classified as:

- *Implanted sensors*, where the sensor is located inside the user's body, for example, intracranial stimulation.
- *On-the-body sensors* are the most commonly used biomedical sensors. Some of those sensors, such as EEG or ECG electrodes, require additional gel to decrease contact resistance.
- *Noncontact sensors*, such as optical sensors and cameras that do not require any physical contact with an object. [Chakravarthy02].

The signal-conditioning module performs (usually analog) signal conditioning prior to AD conversion, such as . This module usually does the amplification, transducer excitation, linearization, isolation, or filtering of detected signals.

The A/D converter changes the detected and conditioned voltage into a digital value [Standardization00, Quality00, Enderle00]. The converter is defined by its resolution and sampling frequency. The converted data must be precisely time-stamped to allow later sophisticated analyses [Santori91].

Although most biomedical sensors are specialized in processing of certain signals, it is possible to use generic measurement components, such as data acquisition (DAQ), or image acquisition (IMAQ) boards, which may be applied to broader class of signals. Creating generic measuring board, and incorporating the most important components of different sensors into one unit, it is possible to perform the functions of many medical instruments on the same computer [Loob00].

3.2. Sensor interface

There are many interfaces used for communication between sensors modules and the computer [Arpia01]. According to the type of connection, sensor interfaces can be classified as *wired* and *wireless*.

- *Wired Interfaces* are usually standard parallel interfaces, such as General Purpose Interface Bus (GPIB), Small Computer Systems Interface (SCSI), system buses (PCI eXtension for Instrumentation PXI or VME Extensions for Instrumentation (VXI), or serial buses (RS232 or USB interfaces) [Tracht93].

- *Wireless Interfaces* are increasingly used because of convenience. Typical interfaces include 802.11 family of standards, Bluetooth, or GPRS/GSM interface [Bhagwat01, Tennenhouse96, Ferrigno02]. Wireless communication is especially important for implanted sensors where cable connection is impractical or not possible [Schwiebert01]. In addition, standards, such as Bluetooth, define a self-identification protocol, allowing the network to configure dynamically and describe itself. In this way, it is possible to reduce installation cost and create plug-and-play like networks of sensors [Dunbar01]. Device miniaturization allowed development of Personal Area Networks (PANs) of intelligent sensors [Jovanov00, Jovanov01b]

Communication with medical devices is also standardized with the IEEE 1073 family of standards [IEEE1073]. This interface is intended to be highly robust in an environment where devices are frequently connected to and disconnected from the network [IEEE00, IEEE98].

3.3. Processing Module

Integration of the general purpose microprocessors/microcontrollers allowed flexible implementation of sophisticated processing functions. As the functionality of a virtual instrument depends very little on dedicated hardware, which principally does not perform any complex processing, functionality and appearance of the virtual instrument may be completely changed utilizing different processing functions.

Broadly speaking, processing function used in virtual instrumentation may be classified as *analytic processing* and *artificial intelligence techniques*.

3.3.1. Analytic processing

Analytic functions define clear functional relations among input parameters. Some of the common analyses used in virtual instrumentation include spectral analysis, filtering, windowing, transforms, peak detection, or curve fitting [Bruse00, Akay97]. Virtual instruments often use various statistics function, such as, random assignment and biostatistical analyses [Dunn00]. Most of those functions can nowadays be performed in real-time.

3.3.2. Artificial intelligence techniques

Artificial intelligence technologies could be used to enhance and improve the efficiency, the capability, and the features of instrumentation in application areas related to measurement, system identification, and control [Piuri98, Spoelder r96, Amigoni02]. These techniques exploit the advanced computational capabilities of modern computing systems to manipulate the sampled input signals and extract the desired measurements.

Artificial intelligence technologies, such as neural networks, fuzzy logic and expert systems, were applied in various applications, including sensor fusion to high-level sensors, system identification, prediction, system control, complex measurement procedures, calibration, and instrument fault detection and isolation [Bernieri95, Piuri98]. Various nonlinear signal processing, including fuzzy logic and neural networks, are also common tools in analysis of biomedical signals [Akay00a, Hudson99, Akay00b].

Using artificial intelligence it is even possible to add medical intelligence to ordinary user interface devices. For example, several artificial intelligence techniques, such as pattern recognition and machine learning, were used in a software-based visual-field testing system [Liu98].

3.4. Database interface

Computerized instrumentation allows measured data to be stored for off-line processing, or to keep records as a part of the patient record [Kilman97, Antony97]. There are several currently available database technologies that can be used for this purpose (Table #).

Table #. *The most frequently used contemporary databases interfaces.*

Database interface	**Description**
File System	Random writing and reading of files.
eXtensible Markup Language (XML)	Standardized markup files.

Open Database Connectivity (ODBC)	SQL based interface for relation databases.
Java Database Connectivity (JDBC)	Java programs' SQL based object-oriented interface for relation databases.
ActiveX Data Objects (ADO)	Windows programs' object-based interface for various data sources including relational databases and XML files.
Data Access Objects (DAO)	Windows programs' object-based interface for relation databases.

Simple usage of file systems interface leads to creation of many proprietary formats, so the interoperability may be a problem. The eXtensible Markup Language (XML) may be used to solve interoperability problem by providing universal syntax [Roy00]. The XML is a standard for describing document structure and content [Bergholz00, W3C]. It organizes data using markup tags, creating self-describing documents, as tags describe the information it contains. Contemporary database management systems such SQL Server and Oracle support XML import and export of data [Lear99].

Many virtual instruments use DataBase Management Systems (DBMSs) [Olansen01]. They provide efficient management of data and standardized insertion, update, deletion and selection. Most of these DBMSs provided Structured Query Language (SQL) interface, enabling transparent execution of the same programs over database from different vendors. Virtual instruments use these DMBSs using some of programming interfaces, such as ODBC, JDBC, ADO, and DAO [LabVIEW].

3.5. Medical information system interface

Virtual instruments are increasingly integrated with other medical information systems, such as hospital information systems. They can be used to create executive dashboards, supporting decision support, real-time alerts and predictive warnings [Olansen01]. Some virtual interfaces toolkits, such as LabView, provide mechanisms for customized components, such as ActiveX objects [Ni-LabView]. That allows communication with other information system, hiding details of the communication from virtual interface code.

In Web based telemedical applications this integration is usually implemented using Unified Resource Locators (URLs). Each virtual instrument is identified with its URL, receiving configuration settings via parameters. The virtual instrument then can store the results of the processing into a database identified with its URL [Jovanov99, Marovic98].

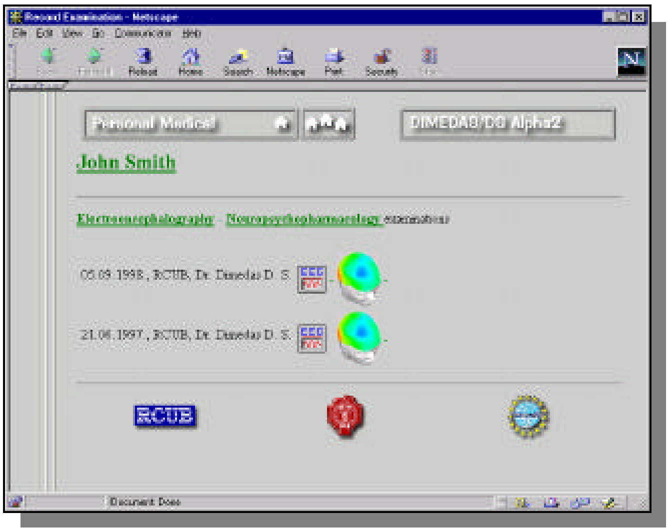

Figure #. *EEG VMDs in DIMEDAS information system; two VMDs are associated with every EEG recording.*

In addition to described integration mechanisms, there are standard for communications among medical applications. For example, OMG Healthcare DTF (http://healthcare.omg.org/), is defining standards and interfaces for healthcare objects, such as CORBAmed standard, in order to develop interoperability technologies for the global healthcare community [Filman01]. Although these standards are still not widely used, they have potential to ensure interoperability among virtual instruments and medical information systems on various platforms.

3.6. Presentation and control

An effective user interface for presentation and control of a virtual instrument affects efficiency and precision of an operator do the measurements and facilitates result interpretation. Since computer's user interfaces are much easier shaped and changed than conventional instrument's user interfaces, it is possible to employ more presentation effects and to customize the interface for each user. According to presentation and interaction capabilities, we can classify interfaces used in virtual instrumentation in four groups:

- terminal user interfaces,
- graphical user interfaces,
- multimodal user interfaces, and
- virtual and augmented reality interfaces.

3.6.1. Terminal User Interfaces

First programs for instrumentation control and data acquisition had character-oriented terminal user interfaces. This was necessary as earlier general-purpose computers were not capable of presenting complex graphics. As terminal user interfaces require little of system resources, they were implemented on many platforms.

In this interfaces, communication between a user and a computer is purely textual. The user sends requests to the computer typing commands, and receives response in a form of textual messages. Presentation is usually done on a screen with fixed resolution, for example 25 rows and 80 columns on ordinary PC, where each cell presents one of the characters from a fixed character set, such as the ASCII set. Additional effects, such as text and background color or blinking, are possible on most terminal user interfaces. Even with the limited set of characters, more sophisticated effects in a form of character graphics are possible.

Although terminal user interfaces are not any more widely use on desktop PCs, they have again become important in a wide range of new pervasive devices, such as cellular phones or low-end personal digital assistants (PDAs). As textual services, such as SMS, require small presentation and network resources they are broadly supported and available on almost all cellular phone devices. These services may be very important in distributed virtual instrumentation, and for emergency alerts [Obrenovic02a].

3.6.2. Graphical User Interfaces (GUI)

Graphical user interfaces (GUIs) enabled more intuitive human-computer interaction, making virtual instrumentation more accessible [Santori91]. Simplicity of interaction and high intuitiveness of graphical user interface operations made possible creation of user-friendlier virtual instruments. GUIs allowed creation of many sophisticated graphical widgets such as graphs, charts, tables, gauges, or meters, which can easily be created with many user interface tools (Figure #). In addition, improvements in presentation capabilities of personal computers allowed for development of various sophisticated 2-D and 3-D medical imaging technologies [Robb99, Webb02, Robb94].

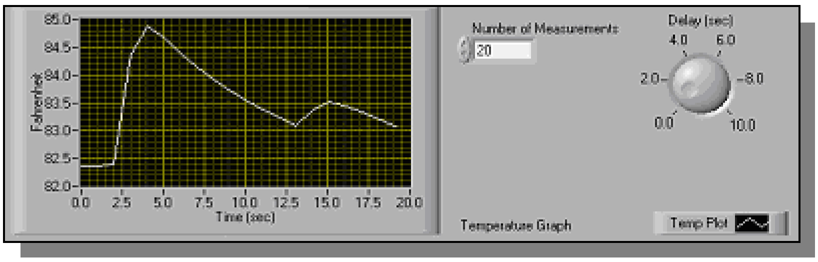

Figure #: *An example virtual instrument graphical user interface.*

Computer graphics extended the functionality of conventional medical diagnostic imaging in many ways, for example, by adding the visual tool of color. For instance, interpretation of radiographs, which are black-and-white images, requires lots of training, but, with color, it is possible to highlight problems clearly [Loob00].

In addition, improvements in presentation capabilities of personal computers allowed for development of various sophisticated 2-D and 3-D medical imaging technologies [Robb99, Webb02, Robb94].

3.6.3. Multimodal presentation

In addition to graphical user interfaces that improve visualization, contemporary personal computers are capable of presenting other modalities such as sonification or haptic rendering. Multimodal combinations of complementary modalities can greatly improve the perceptual quality of user interfaces [Jovanov01a, Oviatt99, Oviatt00, Turk00].

Sonification is the second most important presentation modality. Relationship between visualization and sonification is itself a complex design problem, due to the nature of the cognitive information processing. Efficiency of sonification, as acoustic presentation modality, depends on other presentation modalities. Sonification is effectively used in various biomedical applications, for example, in virtual instruments for EEG analysis [Jovanov01a].

Although not widely available, haptic rendering may be very important upcoming presentation modality for virtual instruments, as the physical contact between a physician and a patient is part of standard examination procedures. There are commercially available haptic interfaces that can relay resistance at about 1000 Hz, and that were used in various surgical simulations [Sorid00].

3.6.4. Virtual and augmented reality

Virtual environments will most likely pervade the medical practice of the future [Akay01]. Many of the goals of virtual reality technology developers actually mirror those involved in virtual instrumentation work [Loob00]. Although virtual reality systems do not necessarily involve the use of virtual instrumentation, they nonetheless drive the development of new conditions under which physicians will need access to data in radically different forms [Sorid00].

A combination of virtual presentation with real world objects creates augment reality interfaces [Rosenbloom02, Poupyerev02]. For example, augmented reality may allow computer generated tumor image from MRI recording to be superimposed on the real view of the patient during surgery.

3.7. Functional Integration

Functional integration of modules governs flexibility of a virtual instrument. The simplest, and the least flexible way, is to create a virtual instrument as a single, monolithic application with all software modules of the virtual instruments logically and physically integrated. This approach can achieve the best performance, but makes difficult maintenance and customization. Therefore, it is more convenient to use modular organization. An object-oriented method [Booch94] was identified as natural approach in modeling and design of instruments [Qingping98] [Daponte92] [Bhaskar86]. Each module of a virtual instrument is then implemented as an object with clearly defined interface, integrated with other objects using message interchange. Similar approach is component-oriented approach, where, in addition to logical separation of components into objects, they are physically placed into different unit to allow reuse [Kozaczynski98].

Another approach, similar in its basic idea to the object-oriented approach, is a structural coupling paradigm for non-conventional controllers that defines layered approach to functional integration of sensor modules [McMillan97]. This sensor model was applied in many domains, including electrophysiological interaction systems with sensors for human physiological signals [Allanson02]. In this sensor interaction model, a stream of raw data from the sensing hardware, for example electroencephalogram (EEG) data, passes through up to two levels of signal preprocessing before it is either passed to an application or presented directly to a subject (Figure #). Second command layer, which is optional, allows more flexible organization of data processing and plug-and-play like integration of complex processing mechanisms into a virtual instrument solution.

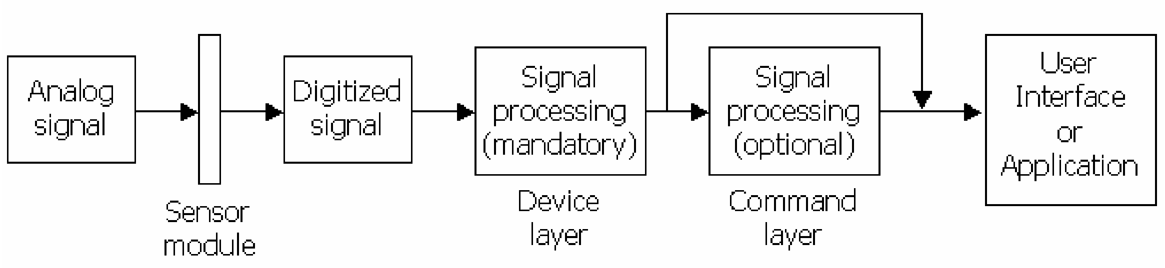

Figure #. *Sensory model described in [McMillan97] and [Allanson02].*

4. Distributed Virtual Instrumentation

Advances in telecommunications and network technologies made possible physical distribution of virtual instrument components into telemedical systems to provide medical information and services at a distance [Huston00, Perednia95]. Distributed virtual instruments is naturally integrated into telemedical systems [Starcevic00][Caldwell98]. Figure # illustrates possible infrastructure for distributed virtual instrumentation, where the interface between the components can be balanced for price and performance [Goldberg00].

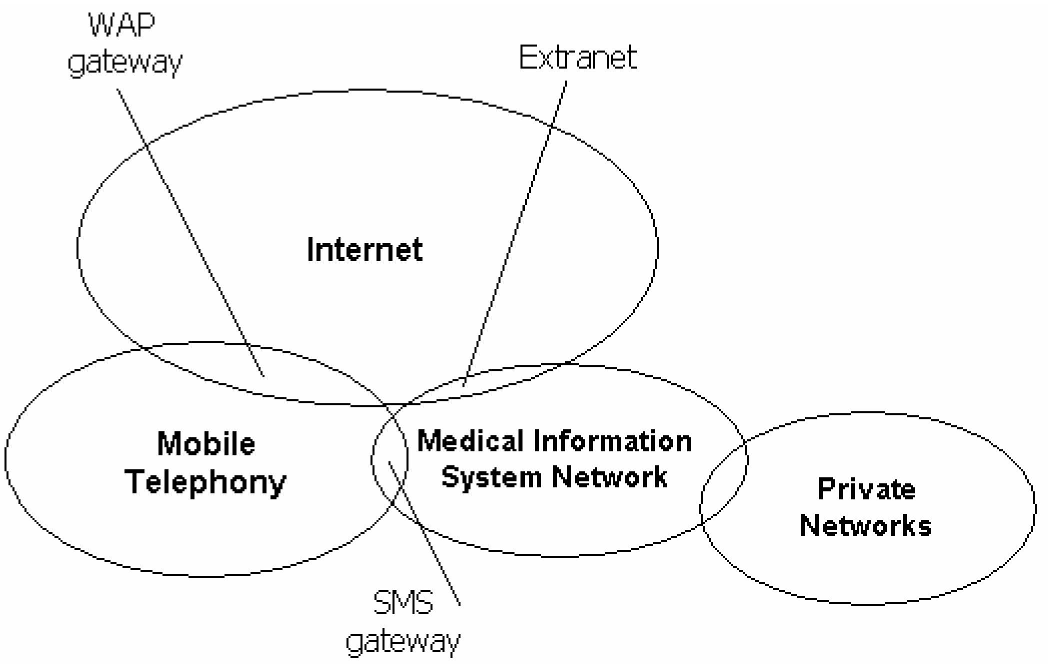

Figure #. *Infrastructure for distributed virtual instrumentation.*

4.1. Medical Information System Networks and Private Networks

Medical information systems, such as hospital information systems, are usually integrated as Intranets using Local Area Network (LAN). Historically, medical information systems were firstly interconnected using various private networks, starting form point-to-point communication with fax and modems connected to analog telephone lines operating at speeds up to 56Kbps, ISDN lines of up to 128Kbps, T-1 lines having a capacity of 1.544Mbps, and satellite links of 100Mbps.

Advanced virtual instrumentation solutions could be implemented using existing local and private networks [Fortino99]. For example, the EVAC project demonstrated a prototype system for using virtual environments to control remote instrumentation, illustrating the potential of a virtual laboratory over high-speed networks [Potter96]. Although private networks improve the performance, reliability and security, they are usually very expensive to develop and maintain.

4.2. The Internet

The Internet has enormous potential for distributed biomedical virtual instrumentation. Various remote devices, such as telerobots or remote experimental apparatus, can be directly controlled from the Internet [Goldberg00]. There are a great number of research activities that explore how the Internet can be applied to medicine [Filman01]. In addition, many of virtual instrumentation development tools, such as LabVIEW [Travis00], directly support integration of virtual instruments in the Internet environment [Jovanov99, Starcevic00, Obrenovic00]. The Web technologies make possible creation of sophisticated client-server applications on various platforms, using interoperable technologies such as HTML, Java applets, Virtual Reality Modeling Language [VRML97], and multimedia support [Jovanov99].

Although the Internet is already enabling technology for many biomedical applications, a recent United States study of health-care applications in relation to Internet capabilities found clear requirements for the Internet's evolutionary development [Davie01]. More serious use of the Internet in clinical applications could be achieved only if the level of service can be guaranteed, including a consistent level of bandwidth end-to-end as well as high reliability and security [Huston00]. However, the rapid progress of the Internet will probably very soon enable its usage in complex real-time applications.

4.3. Cellular Networks

Various mobile devices, such as mobile phones or PDAs, are commonplace today. Moreover, the underlying telecommunication infrastructure of these devices, primarily cellular networks, provides sophisticated data services that can be exploited for distributed applications. The most common data service on cellular networks is exchange of simple textual message. Most of mobile phone devices support simple data communication using standard Short Message System (SMS). Although simple, this system allows the various modes of communication for medical applications [Obrenovic02a]:

Wireless Access Protocol (WAP) is platform independent wireless technology, which enables mobile devices to effectively access Internet content and services, as well as to communicate with each other [Leavitt00]. WAP manages communication by exchanging messages written in Wireless Markup Language (WML). The WAP and the Internet can support new kinds of applications, such as remote monitoring using a wireless personal monitor and cellular phone link connected on request in the case of medical emergencies [Jovanov00, Jovanov01b]. The interface allows the following modes of communications [Obrenovic02a]:

- Emergency WAP push, which sends WML messages to physicians or medical call center in case of medical emergency;
- WML browsing, which allows a participant to browse through information in medical information systems or in monitoring system;
- Data distribution WAP, which periodically sends messages to physicians. These data could be simple text or some 2D graphics with wireless bitmap (WBMP).

4.4. Distributed Integration

When the components are distributed, efficient communication mechanisms are needed. According to a conceptual model and abstractions they utilize, we can identify four approaches to distributed communication:

- Message passing systems,
- Remote procedure calling (RPC) systems,
- Distributed object systems, and
- Agent-based systems.

The message passing model allows communication between programs by exchange of messages or packets over the network. It supports a variety of communication patterns, such as pier-to-pier, group, broadcast and collective communication [Pianegian02]. For example, in virtual instrumentation application the data acquisition part could be a server for other units, sending messages with measured data at request or periodically to processing clients. Data processing clients may themselves be servers for data presentation devices. In a distributed environment there may be many interconnected servers and clients, each dedicated to one of the virtual instrument functions [Grimaldi98].

Remote procedure call (RPC) is an abstraction on top of message passing architectures [RPC]. RPC brings procedural programming paradigm to network programming, adding the abstraction of the function call to distributed systems. In RPC, communication between programs is accomplished by calling a function on that other computer's machine creating the illusion that communication with a remote program is not different than communication with a local program.

Distributed object systems extend the idea of RPC with the object-oriented abstraction on top of procedure calls. Distributed object systems supply programs with references to remote objects, allowing the program to control, call methods, and store the remote object in the same way as a local object. The major standard in distributed objects is OMG CORBA, a language-neutral specification for communicating object systems [CORBA]. Many standards have been defined on top of CORBA, such as *CORBAmed* that defines standardized interfaces for healthcare objects [CORBAmed]. Competitors to CORBA include Microsoft's DCOM architecture [Sessions97] and the various distributed object systems layered on top of Java [RMI].

Agent based integration is potentially very effective distributed virtual instrument integration mechanism. Agent based systems add concepts of autonomity and proactivity to distributed object systems. Agent-oriented approach is well suited for developing complex, distributed systems [Jennings01, Wooldrige99, Wijata00]. Agents can react asynchronously and autonomously to unexpected situations, increasing robustness and fault-tolerance that is very important in the case of fragile network connections, and for mobile devices [Lange99]. As an example of an agent-based distributed integration, we can present a Virtual Medical Device (VMD) agent framework with four types of agents: *data agents*, *processing agents*, *presentation agents*, and *monitoring agents* for distributed EEG monitoring [Obrenovic02b]. In this framework, data agents abstract data source, creating uniform view on different types of data, independent of data acquisition device. Processing agents produce derived data, such us power spectrum from raw data provided by the data agents. Presentation agents supply user interface components using a variety of user data views. User interface components are based on HTTP, SMS and WAP protocols. Monitoring agents collaborate with data and processing agents providing support for data mining operations, and search for relevant patterns.

5. Tools and Platforms

Virtual instrumentation mostly use general-purpose computing equipment, system software and some specialized software modules. In this section we describe hardware platforms, operating systems and development environments often used in development of virtual instruments.

5.1. Hardware Platforms and Operating Systems

Virtual instrumentation and "measurement revolution" is a direct result of another revolution – the PC revolution [NI99] providing the common hardware platform for virtual instrumentation based on Industry Standard Architecture (ISA). However, other personal computing architectures were used too. For example, LabVIEW 1.0 was developed on Macintosh computer, and it still supported by LabVIEW. In addition to desktop personal computers, there are more and more pervasive devices such as Internet-enabled cellular phones, Personal Digital Assistants (PDAs), laptops and wearable PCs. Although still not very often used in virtual instrumentation, these devices are continually evolving, approaching the capabilities of its desktop counterparts. Therefore, these pervasive devices are more and more interesting implementation platform, especially for distributed virtual instrumentation.

Operating systems provide a uniform view on the underlying hardware through device driver layer, isolating details of the sensor interface or sensor device. Commonly used operating systems are: Windows operating systems (MS DOS in early days of virtual instrumentation, 95/98/NT/2000/XP/CE), UNIX/Linux, and MacOS.

5.2. Development environments

Development of virtual instrument is primarily concerned with the development of software, as sensors and hardware are generally available in the open market. We describe two types of virtual instrumentation development environments:

- conventional programming language environments, and
- graphical programming environments.

5.2.1. Programming language environments

As any other program, software for virtual instrument may be developed with any of available general purpose programming environments. In the late 1970s and early 1980s BASIC had been the dominant language used with dedicated instrument controllers [Santori91]. In the mid and late 1980s, new programming languages became common, particularly C, as they allowed high-level programming with very efficient code. The first version of LabVIEW had been written in C.

In early days, a virtual instrument developer had to take care of much of the low-level details, such as work with communication resources and memory management. Today, as the operating system encapsulates underlying hardware providing standardized application programming interfaces (APIs), such as communication APIs or GUI APIs, developers may concentrate on the functional logic of a virtual instrument. Therefore, any programming language that can make use of these APIs, can be used for development, for instance, Visual Basic, Visual C++, Delphi or Java.

In addition to built-in support of programming languages, developers of virtual instruments often use various third-party software libraries, which in many cases are freely available on many platforms. For example, the FFTW is the open-source multi-platform library for efficient FFT analysis [Frigo98]. Another example is the OpenGL, which allows efficient multi-platform development of effective 3D graphics presentations.

Java has been very popular implementation environment for various medical and virtual instrumentation solutions because of its architecture and platform independence [Jepsen00]. Various Java toolkits and virtual instrument environments are available [Allanson02, Grimaldi98, Zubillaga-Elorza99, Zubillaga-Elorza98]. Java programs are first compiled into an intermediate form called bytecode; then they are interpreted at runtime in a platform-specific Java virtual machine (JVM). This approach allows programmers to write Java programs independent of the execution environment, so programs written in Java execute in nearly identical fashion in any Java-aware environment without the need for porting. Java has been used in various medical applications. For example, Java-based medical information systems are used to integrate legacy applications for patient records, billing, and pharmacy that are compatible with the industry standard Health Level 7

(HL7) data interchange format. Other applications developed in Java enable information sharing between healthcare providers and insurance companies. Java supports all aspects of development of virtual instruments, including work with communication resources, files, databases, Internet communication, multimedia, as well as 2D and 3D graphics. Java is also used as a script language for VRML virtual environments [VRML97].

5.2.2. Graphical programming tools

Programming environments described in previous sections, require from designers and users programming proficiency. New generation of graphical programming tools allows system integration for ordinary users. Here we describe two graphical programming tools: Laboratory Virtual Instrument Engineering Workbench (LabVIEW) and BioBench.

Laboratory Virtual Instrument Engineering Workbench (LabVIEW)

National Instruments' LabVIEW made development of virtual instruments more accessible to laboratory users and physicians [Ertugrul02, Khalid01]. LabVIEW is the most popular virtual instrumentation tool, and has been applied to many fields, including virtual bio-instrumentation [Olansen01].

LabVIEW 1.0 was launched in 1986, with the goal of providing a software tool that empowered engineers to develop customized systems [Vose86, Santori91]. LabVIEW is a program development environment, much like Java, C or BASIC. However, while other programming systems use text-based languages to create code, LabVIEW uses a graphical programming language, called G. In LabVIEW programs are formed as block diagrams. LabVIEW uses data-flow programming model, where the execution order is determined by the flow of data between blocks. LabVIEW is also a multitasking and multithreading system.

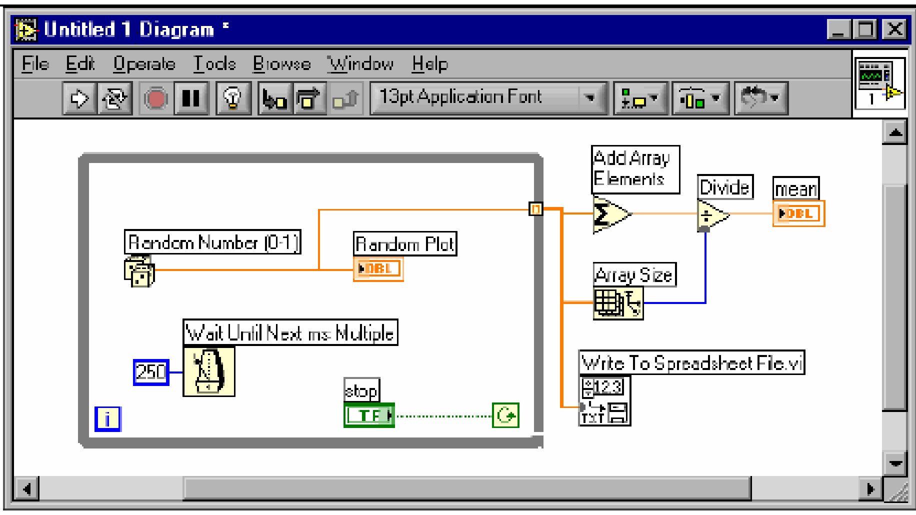

Figure #. *An example of LabVIEW graphical programming notation.*

LabVIEW is a general-purpose programming system with extensive libraries of functions for any programming task. In addition, LabVIEW includes libraries for data acquisition, instrument control, data analysis, data presentation, and data storage. LabVIEW also includes conventional program development tools, such as a debugger and supports vast number of devices and interface standards, has more than 4000 built-in analysis, math, and signal processing functions, as well as support for SQL and ADO database connectivity, and open connectivity through XML, TCP/IP, wireless, and other standards [LabVIEW, Travis00].

BioBench

BioBench is developed as an extension of LabVIEW for biomedical measurements to simplify development of biomedical virtual instruments. LabVIEW greatly simplifies programming by introducing the graphical notation, but still requires a lot of effort to create a virtual instrument. BioBench is primarily designed for physiological data acquisition and analysis, for use in research and academic environments [Olansen01]. It is developed by Premise in collaboration with National Instruments. To execute, the BioBench requires LabVIEW RunTime Engine.

BioBench inherits graphical programming capabilities of LabVIEW while adding customized controls adapted for the measurements of physiological signals such as EEG, ECG or EMG.

6. Biomedical Applications of Virtual Instrumentation

Virtual instrumentation is being increasingly accepted in biomedical field. In relation to the role of a virtual instrument, we may broadly classify biomedical applications of virtual instrumentation in four categories (Figure #):

- *Examination*, where a physician does online or off-line examination of patient measurements,
- *Monitoring*, which can be used as a basis for real-time alerts and interactive alarms,
- *Biofeedback*, where measured signals are presented back to a patient in real-time, and
- *Training and education*, where a virtual instrument may simulate or playback earlier measured signals.

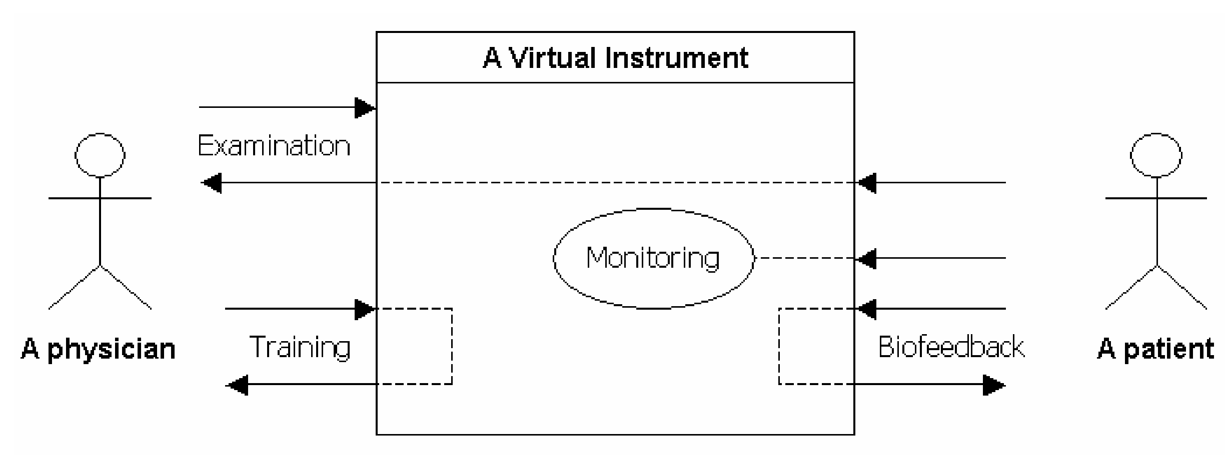

Figure #. *Types of biomedical applications of virtual instrumentation.*

6.1. Examination

Examination systems are open-loop systems that detect biomedical information from a patient and present it to a physician. During examination a physician performs various online or offline analysis of measured patient data in order to make a diagnosis [Parvis02]. Examination can be made locally, in direct contact with a patient, or remotely, where a sensor part is on the patient side connected with a physician through a telecommunication network [Starcevic00]. Nowadays, virtual instrumentation solutions are becoming a part of standard medical examination procedures, with medical systems implemented as virtual instruments. An example of virtual EEG analysis instrument is represented in Figure #.

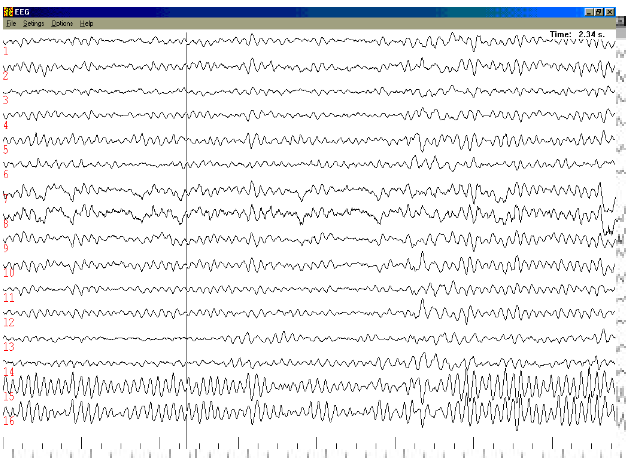

Figure #. *Standard waveform view in the EEG examination virtual instrument [Jovanov99].*

Many active research projects explore biomedical applications of virtual instrumentation [Olansen01], such as canine cardiovascular pressure measurements, cardiopulmonary dynamics measurements or examination of spontaneous cardiac arrhythmia [Fisher95, Rollins00]. Advances in cardiology also make possible design of novel analysis and visualization tools [EMBS02a, EMBS02b].

Some other examples include a virtual instrumentation evaluation system for fiberoptic endoscopes [Rosow97], PC-based noninvasive measurement of the autonomic nervous system used to detect the onset of diabetic autonomic neuropathy [Pruna98], or 3D posture measurement in dental applications [Baszynski02].

6.2. Monitoring

Monitoring is a task in which some process continually tracks the measured data, do some analysis and act upon detection of some pattern. Monitoring systems are also open-loop systems, as the patient is just an object of monitoring. But in contrast to examination systems they are more autonomous. Design of monitoring systems is by itself a complex process, as many real-time requirements have to be fulfilled [Sachenko02, Platonov02]. Integrated with a hospital information system, monitoring can be used as a basis for real-time alerts and interactive alarms [Olansen01].

Monitoring and pattern recognition of biomedical signals may be used outside biomedical field, for example, in affective computing. Physiological parameters that are good indicators of excitement, such as skin conductance and heart rate, are integral data sources for emotional-state-related interactive computer systems [Allanson02].

6.3. Training and education

Virtual instrumentation offers great possibilities for education and improving the skills of physicians. Computer generated models allow education and training of operator without actual sensors, which can greatly reduce cost and duration of training [Schmalzel98, ,Waller00, Jan99, Adam96]. As the same virtual instrument can work online, playback earlier measured data, or simulate any clinical situation, the training experience may not differ significantly from the real-world measurements [Akay01].

Virtual instrumentation may also be integrated with many virtual reality based applications for education and training. For example, Hofman et al. developed VisualizeR, a virtual environment designed to support the teaching and learning of subjects that require understanding of complex 3D structures, such as human anatomy [Hoffman01].

6.4. Biofeedback

Biofeedback systems are closed-loop systems that detect biomedical changes and presented them back to the patient in real time to facilitate change of user's state. For example, physical rehabilitation biofeedback systems can amplify weak muscle signals, encouraging patients to persist when there is a physical response to therapy that is generally not visible [Allanson02]. Interfaces in existing biofeedback applications range from interactive 2D graphical tasks—in which muscle signals, for example, are amplified and transformed into control tasks such as lifting a virtual object or typing, to real-world physical tasks such as manipulating radio-controlled toys [Charles99].

Figure # shows a multimodal interface for simple EEG-based biofeedback system [Obrenovic00]. A position of the pointer on a display is a function of subject's EEG signals. Prior to a training session, a physician records patient's EEG that represents the referential state. Later on, the difference between patient's current state and the pre-recorded state is shown on a display as a pointer deviation. The subjects are trained to move the needle or to keep it on some value.

Figure. *A multimodal interface for simple EEG-based biofeedback system.*

Healthcare providers are increasingly using brain-wave biofeedback or neurofeedback as part of the treatment of a growing range of psychophysiological disorders such attention deficit/hyperactivity disorder (ADHD), post-traumatic stress disorder, addictions, anxiety, and depression. In these applications, surface mounted electrodes detect the brain's electrical activity, and the resulting electroencephalogram (EEG) is presented in real time as abstract images. Using this data in reward/response-based control tasks generates increased or reduced activity in different aspects of the EEG spectrum to help ameliorate these psychophysiological disorders [Allanson02, Moran95].

7. Conclusion

Virtual instrumentation brings many advantages over "conventional" instrumentation. Virtual instruments are realized using industry-standard multipurpose components, and they depend very little on dedicated hardware. Generally, virtual instruments are more flexible and scalable as they can be easily reconfigured in software. Moreover, standard interfaces allow seamless integration of virtual instruments in distributed system. Virtual instrumentation significantly decreases the price of an instrument based on mass-produced general-purpose computing platforms and dedicated sensors for a given application. We expect an increased number of hardware and software modules designed for the virtual instrumentation market [Goldberg00]. They will provide building blocks for the next generation of instrumentation and measurement. It would not be surprise if the prefix *virtual* soon disappear, as virtual instrumentation becomes commonplace.

Virtual instrumentation is rapidly entering biomedical field. Many of the general virtual instrumentation concepts may be directly used in biomedical measurements, but biomedical measurements have their own specific features that must be taken into account [Parvis02]. Therefore, although it is widely used in many biomedical solutions, the virtual instrumentation is not common in critical clinical applications. Having in mind complexity of biomedical phenomena, bringing virtual instrumentation closer to critical biomedical applications will require more testing and more extensive list of developed solutions [Loob00]. However, according to the current trend, we will not be waiting long for this to happen.

Bibliography

Adam96 J.S. Adam, E. Rosow, and T. Karselis, "Educational Applications Using Computerized Virtual Instrumentation", Presented at the Association for the Advancement of Medical Instrumentation in Philadelphia, PA, June 4, 1996.

Akay00a M. Akay (Ed.), Nonlinear Biomedical Signal Processing, Volume 1, Fuzzy Logic, Neural Networks, and New Algorithms, Wiley-IEEE Press, 2000.

Akay00b M. Akay (Ed.), Nonlinear Biomedical Signal Processing, Volume 2, Dynamic Analysis and Modeling, Wiley-IEEE Press, 2000, ISBN: 0-7803-6012-5.

Akay01 M. Akay, A. Marsh (Eds.), Information Technologies in Medicine, Volume 1, Medical Simulation and Education, Wiley-IEEE Press, 2001, ISBN: 0-471-38863-7.

Hoffman01 H. Hoffman, M. Murray, R. Curlee, and A. Fritchle, “Anatomic VisualizeR: Teaching and Learning Anatomy with Virtual Reality”, in M. Akay, A. Marsh (Eds.), Information Technologies in Medicine, Volume 1, Medical Simulation and Education, Wiley-IEEE Press, 2001, pp. 205-218.

Akay97 M. Akay (Ed.), Time Frequency and Wavelets in Biomedical Signal Processing, Wiley-IEEE Press, 1997, ISBN: 0-7803-1147-7.

Allanson02 J. Allanson, "Electrophysiologically Interactive Computer Systems", Computer 35(3), 60-65 (2002).

Amigoni02 F. Amigoni, A.Brandolini, G.D'Antona, R.Ottoboni, and M.Somalvico, "Artificial Intelligence in Science of Measurements and the Evolution of the Measurements Instruments: A Perspective Conception", IEEE International Symposium on Virtual Environments and Measurement Systems VIMS'02, May 2002, paper VM-2015.

Antony97 R.T. Antony, "Database support to data fusion automation", Proceedings of the IEEE 85(1), 39 -53 (1997).

Baszynski02 M.Baszynski, C.R.Allen, Z.Moron, P.A.Heasman, "3D Posture Measurement in Dental applications", IEEE International Symposium on Virtual Environments and Measurement Systems VIMS'02, May 2002, paper VM-2017.

Bergholz00 A. Bergholz , "Extending Your Markup: An XML Tutorial", IEEE Internet Computing 4(4), 74-79 (2000).

Bernieri95 A. Bernieri, G. Betta, A. Pietrosanto, and C. Sansone, "A neural network approach to instrument fault detection and isolation", IEEE Transactions on Instrumentation and Measurement 44(3), 747 -750 (1995).

Bhagwat01 P. Bhagwat, "Bluetooth: Technology for Short-Range Wireless Apps", IEEE Internet Computing 5(3), 96-103 (2001).

Bhaskar86 K. S. Bhaskar, J. K. Pecol, and J. L. Beug, "Virtual instruments: object-oriented program synthesis", Conference proceedings on Object-oriented programming systems, languages and applications, ACM Press, New York, NY, USA, 1986, pp. 303-314.

Booch94 G. Booch, Object-Oriented Analysis and Design with Applications, 2nd ed., Benjamin/Cummings, 1994.

Bruse00 E.N. Bruce, Biomedical Signal Processing and Signal Modeling, 2000.

Caldwell98 N.H.M. Caldwell, B.C. Breton, and D.M. Holburn, "Remote Instrument Diagnosis on the Internet", IEEE Intelligent Systems 2(3), 70-76 (1998).

Chakravarthy02 S. Chakravarthy, R. Sharma, and R. Kasturi, "Noncontact level sensing technique using computer vision", IEEE Transactions on Instrumentation and Measurement 51(2), 353-361 (2002).

Charles99 J. Charles, "Neural Interfaces Link the Mind and the Machine", Computer 32(1), 16-18 (1999).

CORBA Object Management Group (OMG). The Common Object Request Broker: Architecture and Specification (CORBA), revision 2.0. Object Management Group (OMG), 2.0 edition.

Daponte92 P. Daponte, L. Nigro, and F. Tisato, "Object oriented design of measurement systems",IEEE Instrumentation & Measurement Magazine 41(6), 874–880 (1992).

Davie01 B. Davie, V. Florance, A. Friede, J. Sheehan, J.E. Sisk, "Bringing Health-Care Applications to the Internet", IEEE Internet Computing 5(3), 42-48 (2001).

Denning01 P.J. Denning, "Origin of Virtual Machines and Other Virtualities", IEEE Annals of the History of Computing 23(3), 73 (2001).

Dunbar01 M. Dunbar, "Plug-and-play sensors in wireless networks", IEEE Instrumentation & Measurement Magazine 4(1), 19-23 (2001).

Dunn00 O.J. Dunn, V.A. Clark, Basic Statistics: A Primer for Biomedical Sciences, 3rd Edition, John Wiley & Sons, Inc., 2000.

EMBS02a J.F. Thayer, "Progress in Heart-Rate Variability Analysis", IEEE Engineering in Medicine and Biology Magazine 21 (4), 22 -23 (2002).

EMBS02b E. Tatara and A. Cinar, "Interpreting ECG data by integrating statistical and artificial intelligence tools", IEEE Engineering in Medicine and Biology Magazine 21(1), 36-41 (2002).

Ertugrul02 N. Ertugrul, LabVIEW for Electric Circuits, Machines, Drives, and Laboratories, Prentice Hall PTR, 2002.

Ferrigno02 L.Ferrigno, A.Pietrosanto, "A Bluetooth-based proposal of instrument wireless interface", IEEE International Symposium on Virtual Environments and Measurement Systems VIMS'02, May 2002, paper VM-2003.

Filman01 R.E. Filman, "Medicine on the Net: From Isolation to Universal Connectivity", IEEE Internet Computing 5(3), 40-41 (2001).

Fisher95 J.P. Fisher, J. Martin, F.P. Day, E. Rosow, J. Adam, C. Chen, and L.D. Gillam, "Validation of a Less Invasive Method for Determining Preload Recruitable Stroke Work Derived with Echocardiographic Automatic Boundary Detection." Circulation 92, 1-278 (1995)

Fortino99 G. Fortino and L. Nigro, "Development of virtual data acquisition systems based on multimedia internetworking", Computer Standards & Interfaces 21(5), 429-440 (1999).

Frigo98 M. Frigo and S. G. Johnson, "FFTW: An Adaptive Software Architecture for the FFT", 1998 ICASSP conference proceedings, Vol. 3, pp. 1381-1384, also http://www.fftw.org/.

Geddes89 L. A. Geddes and L. E. Baker, Principles of Applied Biomedical Instrumentation, 3rd Edition, John Wiley & Sons, 1989.

Goldberg00 H. Goldberg, "What Is Virtual Instrumentation?", IEEE Instrumentation and Measurement Magazine 3(4), 10-13 (2000).

Goldberg00 K. Goldberg, "Introduction: The Unique Phenomenon of a Distance", in

K. Goldberg (ed.), The Robot in the Garden: Telerobotics and Telepistemology in the Age of the Internet, MIT Press, 2000.

Grimaldi98 D. Grimaldi, L. Nigro, and F. Pupo, "Java-based distributed measurement systems", IEEE Transactions on Instrumentation and Measurement 47(1), 100 -103 (1998).

Hudson99 D.L. Hudson and M.E. Cohen, Neural Networks and Artificial Intelligence for Biomedical Engineering, Wiley-IEEE Press, 1999, ISBN: 0-7803-3404-3.

Huston00 T.L. Huston and J.L. Huston, "Is Telemedicine A Practical Reality?", Communications of the ACM 43(6), 91-95 (2000).

IEEE00 IEEE standard for medical device communications - physical layer interface - cable connected, IEEE Std 1073.4.1, IEEE, 2000 Edition, 2000, http://standards.ieee.org/catalog/olis/meddev.html.

IEEE1073 IEEE 1073 family of standards, http://standards.ieee.org/catalog/olis/meddev.html

IEEE98 R.J. Kennelly, "The IEEE 1073 Standard for Medical Device Communications", IEEE Systems Readiness Technology Conference AUTOTESTCON '98, IEEE, 1998, pp. 335 -336.

Jan99 J. Zidek, "Application of Graphical Programming and Benefit of Virtual Instrumentation in Teaching", International Network for Engineering Education and Research Conference - ICEE'99, 1999, Czech Republic, Paper No. 277.

Jennings01 N.R. Jennings, "An Agent-based Approach for Building Complex Software Systems", Communications of the ACM 44(6), 35-41 (2001).

Jepsen00 T. Jepsen, "Don't Know Beans? Try Java", IT Professional 2(2), 29-35 (2000).

Jovanov00 E. Jovanov, J. Price, D. Raskovic, K. Kavi, T. Martin, and R. Adhami, "Wireless Personal Area Networks in Telemedical Environment", Third IEEE EMBS Information Technology Applications in Biomedicine – Workshop of the International Telemedical Information Society ITAB-ITIS 2000, Arlington, Virginia, 2000, pp. 22-27.

Jovanov01a E. Jovanov, D. Starcevic and V. Radivojevic, "Perceptualization of Biomedical Data", in M. Akay, A. Marsh (Eds.), Information Technologies in Medicine, Volume 1, Medical Simulation and Education, Wiley-IEEE Press, 2001, pp. 189-204.

Jovanov01b E. Jovanov, D. Raskovic, J. Price, A. Moore, J. Chapman, and A. Krishnamurthy, "Patient Monitoring Using Personal Area Networks of Wireless Intelligent Sensors," 38th Annual Rocky Mountain Bioengineering Symposium, Copper Mountain, Colorado, April 2001.

Jovanov99 E. Jovanov, D. Starcevic, A. Samardžic, A. Marsh, Ž. Obrenovic, "EEG analysis in a telemedical virtual world", Future Generation Computer Systems 15, 255-263 (1999);

Khalid01 S.F. Khalid and J. Laney, Advanced Topics in LabWindows/CVI, Prentice Hall PTR, 2001.

Kilman97 D.G. Kilman, D.W. Forslund, "An International Collaboratory Based on Virtual Patient Records", Communications of the ACM, 40(8), 111-117 (1997).

Kozaczynski98 W. Kozaczynski, G. Booch, "Component-Based Software Engineering", IEEE Software 15(5), 34-36 (1998).

LabVIEW NI LabVIEW Web Site, http://ni.com/info

Lange99 D.B. Lange, M. Ochima, "Seven Good Reasons for Mobile Agents", Communications of the ACM 42(3), 88-89 (1999).

Lear99 A.C. Lear , "XML Seen as Integral to Application Integration", IT Professional 1(4), 12-16 (1999).

Leavitt00 N. Leavitt, "Will WAP Deliver the Wireless Internet?", Computer 33(5), 16-20 (2000).

Liu98 X.Liu, G.Cheng, and J.X. Wu, "AI for Public Health: Self-Screening for Eye Diseases", IEEE Intelligent Systems 13(5), 28-35 (1998).

Loob00 W. Loob, "Virtual Instrumentation Boosts Capabilities and Reduces Costs", Medical Device & Diagnostic Industry Magazine, September 2000, pp. 10.

Marovic98 B. Marovic and Z. Jovanovic, "Visualisation of 3D fields and medical data using VRML", Future Generation Computer Systems 14 (1–2), 33–49 (1998).

McMillan97 G.R. McMillan, R.G. Eggleston, and T.R. Anderson, "Nonconventional Controls", in G. Salvendy, ed., Handbook of Human Factors and Ergonomics, 2nd ed., John Wiley & Sons, New York, 1997, pp. 729-771.

Moran95 P.T. Moran, G.F. Harris, K. Acharya, Z. Hongsheng, and J.J. Wertsch, "A biofeedback cane system: instrumentation and subject application results", IEEE Transactions on Rehabilitation Engineering 3(1), 132 -138 (1995).

Nebeker02 F. Nebeker, "Golden Accomplishments in Biomedical Engineering", IEEE Engineering in Medicine and Biology Magazine 21(3), 17-47 (2002).

NI99 National Instruments, "The Measurement Revolution, White Paper", September 1999, http://www.ni.com/revolution/learn.htm.

Obrenovic00 Z. Obrenovic, D. Starcevic, E. Jovanov, and V. Radivojevic, "An Implementation of Real-time Monitoring and Analysis in Telemedicine", Third IEEE EMBS Information Technology Applications in Biomedicine – Workshop of the International Telemedical Information Society ITAB-ITIS 2000, Arlington, Virginia, 2000, pp. 74-78.

Obrenovic02a Z. Obrenovic, D. Starcevic, E. Jovanov, and V. Radivojevic, "Implementation of Virtual Medical Devices in Internet and Wireless Cellular Networks", In Wojciech Cellary, Arun Iyengar (Eds.): Internet Technologies, Applications and Social Impact, IFIP TC6/WG6.4 Workshop on Internet Technologies, Applications and Social Impact (WITASI 2002), IFIP Conference Proceedings 232, Kluwer, 2002, 229-242.

Obrenovic02b Z. Obrenovic, D. Starcevic, E. Jovanov, V. Radivojevic, "An Agent Based Framework for Virtual Medical Devices", Proceedings of The First International Joint Conference on Autonomous Agents and Multiagent Systems AAMAS 2002, ACM Press, New York, NY, USA, 2002, 659-660.

Olansen01 J.B. Olansen and E.Rosow, Virtual Bio-Instrumentation: Biomedical, Clinical, and Healthcare Applications in LabVIEW, Prentice Hall Inc., 2001.

Oviatt00 S. Oviatt and P. Cohen, "Perceptual user interfaces: multimodal interfaces that process what comes naturally", Communications of the ACM 43(3), 45-53 (2000).

Oviatt99 S. Oviatt, "Ten Myths of Multimodal Interaction", Communications of the ACM 42(11), 74-81 (1999).

Parvis02 M. Parvis and A. Vallan, "Medical measurements and uncertainties", IEEE Instrumentation & Measurement Magazine 5(2), 12 -17 (2002).

Perednia95 T.L. Huston and J.L. Huston, "Is Telemedicine A Practical Reality?", Communications of the ACM 43(6), 91-95 (2000).

Pianegian02 P.Pianegiani, D.Macii, and P.Carbone, "Open Distributed Control and Measurement System Based on an Abstract Client-Server Architecture", IEEE International Symposium on Virtual Environments and Measurement Systems VIMS'02, May 2002, paper VM-2007.

Piuri98 C. Alippi, A. Ferrero, and V. Piuri, "Artificial intelligence for instruments and measurement applications", IEEE Instrumentation & Measurement Magazine 1(2), 9 -17 (1998).

Platonov02 A.A. Platonov and W. Winiecki, "Statistical synthesis and optimal decomposition in intelligent monitoring systems design", Computer Standards & Interfaces 24(2), 101-110 (2002).

Potter96 C. Potter, R. Brady, P. Moran, C. Gregory, B. Carragher, N. Kisseberth, J. Lyding, and J. Lindquist, "EVAC: a virtual environment for control of remote imaging instrumentation", IEEE Computer Graphics and Applications 16(4), 62 -66 (1996).

Poupyerev02 I. Poupyerev, D.S. Tan, M.Billinsghurst, H.Kato, H.Regenbrecht, N.Tetsutani, "Developing a Generic Augmented-Reality Interface", Computer 35(3), 44-50 (2002).

Pruna98 S. Pruna and A. Dumitrescu, "PC-based noninvasive measurement of the autonomic nervous system. Detecting the onset of diabetic autonomic neuropathy", IEEE Engineering in Medicine and Biology Magazine 17(2), 66 -72 (1998).

Qingping98 Y. Qingping and C. Butler, "An object-oriented model of measurement systems", IEEE Transactions on Instrumentation and Measurement 47(1), 104 -107 (1998).

Quality00 H. Schumny, "Quality and standardisation aspects of digitalisation", Computer Standards & Interfaces 22(4), 237-238 (2000).

RMI RMI: Remote Method Invocation Web Site. http://java.sun.com/products/jdk/rmi/index.html

Robb94 R.A. Robb, Three-Dimensional Biomedical Imaging: Principles and Practice, John Wiley & Sons, 1994.

Robb99 R.A. Robb, Biomedical Imaging, Visualization, and Analysis, John Wiley & Sons, 1999.

Rollins00 D.L. Rollins, C.R. Killingsworth, G.P. Walcott, R.K. Justice, R.E. Ideker, and W.M. Smith, "A telemetry system for the study of spontaneous cardiac arrhythmias", IEEE Transactions on Biomedical Engineering 47(7), 887 -892 (2000).

Rosenbloom02 A. Rosenbloom, "How the Virtual Inspires the Real", Communications of the ACM 45(7), 29-30 (2002).

Rosow97 E. Rosow, J. Adam, F. Beatrice, K. Youngdahl, and R. Orlando, "A Virtual Instrumentation Evaluation System for Fiberoptic Endoscopes", Presented at the Society of American Gastrointestinal Endoscopic Surgeons (SAGES) March 21, 1997, San Diego, CA.

Roy00 J.Roy and A. Ramanujan, "XML: Data's Universal Language", IT Professional 2(3), 32-36 (2000).

RPC Network Working Group Sun Microsystems. RPC: Remote Procedure Call Protocol Specification. RFC 1057, June 1988. http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1057.txt

Sachenko02	A. Sachenko, "Intelligent data acquisition and advanced computing systems", Computer Standards & Interfaces 24(2), 97-100 (2002).
Santori91	M. Santori, "An instrument that isn't really", IEEE Spectrum 27(8), 36 - 39 (1990).
Schmalzel98	J.L. Schmalzel, "Instrumentation education", IEEE Instrumentation & Measurement Magazine 1(1), 34 -35 (1998).
Schwiebert01	L. Schwiebert , S.K.S. Gupta, J. Weinmann, "Research challenges in wireless networks of biomedical sensors", Proceedings of the seventh annual international conference on Mobile computing and networking, ACM Press, New York, NY, USA, 2001, pp. 151-165.
Sessions97	R. Sessions, COM and DCOM: Microsoft's Vision for Distributed Objects, John Wiley & Sons, 1997.
Sorid00	D. Sorid and S.K. Moore, "The Virtual Surgeon", IEEE Spectrum 37(7), 26-31 (2000).
Spoelder96	F. Russo, "Fuzzy systems in instrumentation: fuzzy signal processing", IEEE Transactions on Instrumentation and Measurement 45(2), 683 -689 (1996).
Spoelder97	H.J.W. Spoelder, A.H. Ullings, and F.C.A. Groen, "Virtual instrumentation: a survey of standards and their interrelation", Conference Record IMTC '97, Vol. 1, 1997, pp. 676-681.
Standardization 00	P. Arpaia , "Introduction to the special issue on Digital instruments standardisation", Computer Standards & Interfaces 22(2), 85-88 (2000).
Starcevic00	D. Starcevic, E. Jovanov, V. Radivojevic, Ž. Obrenovic, A. Samardžic, "Virtual medical devices for telemedical applications", in P. Spasic, I. Milosavljevic, M. Jancic-Zguricas, (Eds.), "Telemedicine", Academy of Medical Sciences of Serbian Medical Association, Belgrade, Yugoslavia, 2000, pp. 218-244.
Tennenhouse96	D.L. Tennenhouse , V.G. Bose, "The SpectrumWare approach to wireless signal processing", Wireless Networks 2(1), (1996)
Tracht93	A.E. Tracht, "Adapting Laptops to Data Acquisition", IEEE Spectrum 30(10), 45-47 (1993).
Travis00	J. Travis, Internet Applications in LabVIEW, Prentice-Hall, 2000.
Turk00	M. Turk and G. Robertson, "Perceptual user interfaces (introduction)", Communications of the ACM 43(3), 33-35 (2000).
Vos97	F.M. Vos, G.L. van der Heijde, H.J.W. Spoelder, I.H.M van Stokkum, and F.C.A. Groen, "A new instrument to measure the shape of the cornea based on pseudorandom color coding", IEEE Transactions on Instrumentation and Measurement 46(4), 794 -797 (1997).
Vose86	M.G. Vose and G. Williams, "Laboratory virtual instrument engineering workbench", BYTE 11(9), (1986).
VRML97	The VRML Consortium Incorporated, Information technology -- Computer graphics and image processing -- The Virtual Reality Modeling Language (VRML) -- Part 1: Functional specification and UTF-8 encoding. International Standard ISO/IEC 14772-1:1997, 1997, http://www.vrml.org/Specifications/VRML97/index.html.
W3C	World Wide Web Consortium Web Site, http://www.w3.org/
Waller00	J.C. Waller, N.Foster, "Training via the web: a virtual instrument", Computers & Education 35(2), 161-167 (2000).
Webb02	A.G. Webb, Introduction to Biomedical Imaging, 2002.

Wijata00 Y.I. Wijata, D. Niehaus, and V.S. Frost, "A scalable agent-based network measurement infrastructure", IEEE Communications Magazine 38(9), 174 -183 (2000).

Wooldrige99 M.J. Wooldrige, N.R. Jennings, "Software Engineering with Agents: Pitfalls and Pratfalls", IEEE Internet Computing 3(3), 20-27 (1999).

Zubillaga-Elorza98 F. Zubillaga-Elorza and C.R. Allen, “A Java-Based Electronic Design Automation Toolkit for the Simulation and Testing of Instruments”, Proceedings of the IEEE Instrumentation and Measurement Technology Conference IMTC '98, IEEE Press, pp. 304–309, 1998.

Zubillaga-Elorza99 F. Zubillaga-Elorza and C.R. Allen, "Virtual Instrument Toolkit: Rapid Prototyping on the Web", IEEE Internet Computing 3(5), 41-48 (1999).