MEDINFO 2007
HISA Ltd Australia
© 2007 The authors. All rights reserved.

Open Source Patient Data Management System for Intensive Care

Massaut J, Reper P, Hooghe L, Gottignies P

Surgical Intensive Care, Brugmann Hospital, Université Libre de Bruxelles

Abstract and objective

In Intensive Care Units, the amount of data to be processed for patients care, the turn over of the patients, the necessity for reliability and for review processes indicate the use of Patient Data Management System (PDMS). To respond to the needs of a Surgical Intensive Care Unit, we developed a PDMS based on open source software and components.

The software was designed as a client–server architecture running on the Linux operating system and powered by the PostgreSQl data base system. The client software was developed in C. The application offers the following functions: medical notes captures, observations and treatments, nursing charts with administration of medications and scoring systems functionalities. The PDMS was used to care more than two thousands patients with the expected reliability and functionalities.

Key words:

database management system, software, intensive care

Introduction

Patient Data Management Systems are mandatory in Intensive Care Unit in response to the amount of data to be processed, the turn over of the patients and the necessity for reliability and review processes. Open Source Software (OSS) by publishing source code allows sharing of software resources and experience. To respond to the needs of our unit and benefit of resources from OSS we developed a PDMS based on open source software and components.

Methods

The software was designed as a client-server architecture running on the Linux operating system (SUSE Linux Enterprise Server 8.0). It uses the PostgreSQL relational database (v 7.2). The client software was developed in C using the GTK interface library. Remote access from remote PCs is implemented by virtual network connections (VNC) : the use of VNC servers on linux servers and VNC viewers on Windows PCs.

The hardware consists in two Intel x86 servers with uninterrupted power supply, one master and one slave to assure the integrity of the database by replication, 14 medical grade panel PCs (Advantek PPC-153m) connected via RS232 medical bus to the patient's monitoring devices and to the servers via dedicated local Intranet network. The master server runs the postgreSQL database, the client-software and the VNC servers. The slave server is used for replication of the database by drdb (Distributed Replicated Block Device). The 14 medical grade panel PCs run the client software at the bedsides and acquire data from the monitoring devices.

The software, developed in C on the Linux platform, offers the following functions:

1. Medical notes captures with patient's history, observations and treatments,
2. Nursing charts for vital signs, IN-OUT balance, ventilation parameters and settings,
3. Functionalities for administration of medications,
4. Scoring system possibilities for patient's classification. (APACHE II, SAPS II, SOFA scores).
5. Reporting at the end of hospitalization in Intensive Care.

Interoperability between these modules is realized through access to the PostgreSQL database and not the use of local memory in the interface. The software was developed to be open source in all its components and is interfaced with Open Office for reporting..

Results

The PDMS was used in our unit from February 2004 for the care of more than two thousands patients. The system is accessible at every bed through panel PCs and at desks or offices through VNC viewers on windows PCs. Its design allowed an access to the database's functionalities with a high availability level (less than 5 hours of interruption over one year).

The use of open source resources was effective to customize the solution to ICU's request and contributed to the acceptability of the software. The use of the C language permitted to obtain small response times but limits the portability of the system and complicated the debugging process in this critical environment. For that reason, Valgrind software was used to systematically track runtimes errors. As pointed before, the system was well accepted locally, but was harder to interface with the information system of the hospital. The PostgreSQL data base largely

contributed to the overall efficacy and robustness of the system.

Discussion

We developed the present software to respond to the needs of our surgical unit, with the hope that this will enhance quality in our unit. In a review of Clinical Informatics in Critical Care, G. Daniel Martich describes several reasons to implement information system in intensive care.[1] The first one is that information systems could reduce medical errors and first of all medications errors. The second reason is that information overload is present at point of care in intensive care units. Clinical informatics at the bedside can help to better manage this load. Other reasons are described like necessity to achieve and assess compliance to guidelines and accreditation rules.

We decided to base our development work on OSS for tree main reasons, first to benefit of the large OSS library and resources, second to avoid to be locked into proprietary software and third to be able to adapt the software to the manual procedures preexisting in our unit. Economical reasons were also present. These reasons are similar to that described by Douglas Carnal. That author described in 2000 that open collaboration over the internet is changing development methods and that OSS will be a significant part of the Medical Software's Future. [2]

Intensive care environments require systems with high availability. The system described here was able to respond to these requirements by the use of dedicated and duplicated servers and the use of dedicated local network for communication between bedside Panel PCs and servers.

Software development and testing for Intensive Care need to achieve high reliability. The C language used to develop the software is unfortunately not by itself a safe language. For that reason, we systematically tested the software with Valgrind, a suite of simulation based debugging and profiling tools, to track run-time errors. The uses of static analysis of the C code with tools like Splint [3] early in the development process and before compilation, or the use of safer languages like Ada or SparkAda [4] are however a better solutions and are used to develop secure systems.

The lack of module specifically designed to communicate with other medical software and applications is a limitation of the system. Development of a communication module with ProGen/HL7 library (an implementation of HL7 in C++) or with Mirth (implemented in java 1.5) would greatly facilitate the integration of the PDMS with other medical software and the hospital's information system.

Conclusion

PDMS based on open source software components are effective and able to respond to the needs of the ICU environment, with a high availability level. The use of OSS allowed us to customize the software to the preexisting organization of the unit and contributed to the acceptability of the whole system. Better integration in the hospital's information system necessitates development of a module specifically designed to communicate with other medical software and applications.

References:

[1] Martich GD, Waldmann CS, Imhoff M. Clinical Informatics in Critical care. J. Intensive Care Med. 2004; 19: 154-63.

[2] Carnall D. Medical software's free future. BMJ 2000; 321: 976.

[3] Evans D, Larochelle D. Improving Security Using Extensible Lightweight Static Analysis. IEEE SOFTWARE, January/February 2002, pp, 42-51.

[4] Hall A. and Chapman R. Correctness by construction: developing a commercial Secure System. IEEE Software January/February 2002, pp, 18-25.