Computer Science • Vol. 8 Special Edition • 2007

Wiesław Wajs[*], Krzysztof Rączka[**],
Paweł Stoch[*], Piotr Kruczek[***]

INTEGRATION PLATFORM AS CENTRAL SERVICE OF DATA REPLICATION IN DISTRIBUTED MEDICAL SYSTEM

The paper presents the application of Java Integration Platform (JIP) to data replication in the distributed medical system. After an introductory part on the medical system's architecture, the focus shifts to a comparison of different approaches that exist with regard to transferring data between the system's components. A description is given of the historical data processing and of the whole area of the JIP application to the medical system.

Keywords: *medical system, data replication, distributed architecture, entity relationship diagram (ERD), XML, Java*

PLATFORMA INTEGRACYJNA JAKO CENTRALNY SERWIS REPLIKACJI DANYCH W SIECIOWYM SYSTEMIE MEDYCZNYM

Artykuł prezentuje wykorzystanie platformy integracyjnej JIP (Java Integration Platform) do realizacji replikacji danych w sieciowym systemie medycznym. Przedstawiono architekturę systemu medycznego, a następnie porównano różne podejścia do przesyłania danych pomiędzy komponentami systemu. Omówiono również przetwarzanie danych historycznych oraz pełny obszar wykorzystania platformy JIP w systemie medycznym.

Słowa kluczowe: *system medyczny, replikacja danych, architektura rozproszona, diagram encji, XML, Java*

1. Introduction

Primary tool used to collect and process data applicable to description of parameters characterizing health condition of infants hospitalized at Ward of Infants Intense Care of Polish-American Institute of Pediatrics at the Medical College of the Jagiellonian University in Cracow is a medical computer system delivered by the Department of

[*] Department of Automatics, AGH University of Science and Technology, Kraków, Poland
`wwa@ia.agh.edu.pl`; `pstoch @ia.agh.edu.pl`
[**] DHL Aviation, Brussels, Belgium `kris.raczka@dhl.com`
[***] Institute of Pediatrics, Jagiellonian University Medical College, Kraków,
`kruczekpiotr@poczta.onet.pl`

Automatics of the Faculty of Electrical Engineering, Automatics, Computer Science and Electronics at the AGH University of Science and Technology. The system joins static medical data, part of which was previously stored in Neonatal Information System (NIS) database, with dynamic data obtained directly from medical devices. The level of complexity causes necessity of storing the data in over 200 tables, and monthly growth of data from only pulse oximeters exceeds 200 MB for one ward of the hospital. The system architecture assumes storing of full data set in central medical database, and partial mirroring of the data in local databases. System characteristics require strong support in the area of data replication, which is entirely realized using Java Integration Platform (JIP).

2. Logical structure

The medical system is a comprehensive solution serving both daily hospitals' wards' needs and also equipped in a dedicated module predicting parameters describing patients' health condition. The system stores data in the following categories:

- reference data,
- prenatal period,
- admission of patients to the ward,
- daily data,
- respiratory sets,
- laboratory tests,
- dynamic data from medical devices,
- discharge of infants from the ward,
- transportation of patients to and from the hospital,
- prediction of health condition of patients.

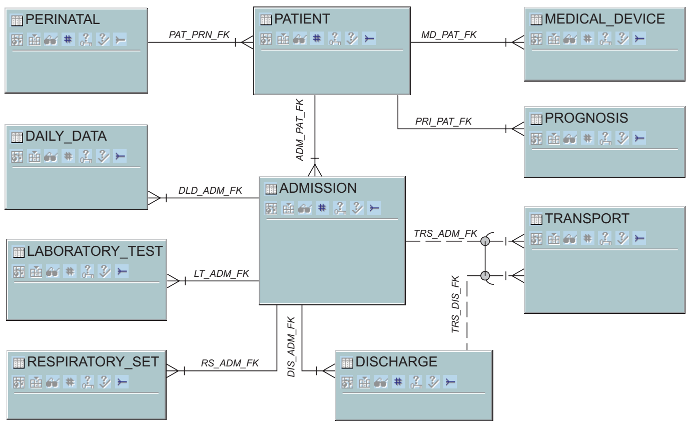

Fig. 1. Simplified entity relationship diagram

The system model has been built in form of an entity relationship diagram, showing well the relational nature of collected data. It uses a wide arsenal of implementation techniques, such as database tables and views, sequences, domains, and also employs commonly data validation methods in form of primary, unique and foreign keys.

The full data model depicts complicated structure of the relationship between entities. The model presented in this document (Fig. 1) is simplified to the level of main concepts.

3. System architecture

Already introduced logical structure of solution together with the specific requirements related to the data availability and security introduces the necessity of use of distributed system architecture, in which the full data set is stored in the module of central database, and specific areas of the data are additionally mirrored locally, directly at the wards of the hospital (Fig. 2). Such architecture got chosen as the best suited for open character of the medical system, giving direct possibility of integration of successive hospital's wards or entire hospitals into the already existing system. The additional advantage of such an approach lays in the fast access to the local database enabling unaffected functioning of application even in case of problems with accessing the central database. Local databases are also utilized for buffering of data collected directly from medical devices. Current version of system is equipped with interfaces enabling communication with pulse oximeters and respirators. Next versions will be extended with graphical visualization of data originating from radiology, tomography, ultrasonography and magnetic resonance.

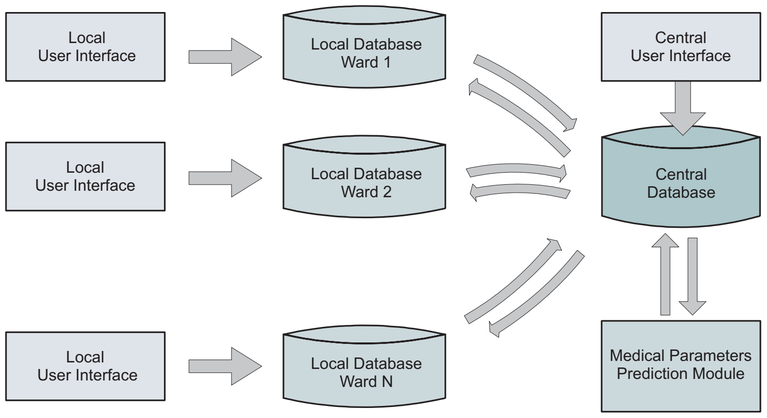

Fig. 2. Global logical system architecture

Medical devices are characterized by a specific structure of data transmission, proprietary for each manufacturer, and by lack of commonly adopted standards. Such situation enforces creation of a dedicated driver for every single device. The task of such a component is to obtain access to the specific data format, decoding of information, optional calculation of derivative data and providing output stream of data in unified format. Taking into consideration continuous character of data stream it is important to employ stable buffer of data, which will guarantee persistence of data even in case of communication problems with remote system components. Out of few existing methods of implementation of data queue the most suitable approach in this case was to introduce local database, which besides of role of dynamic data buffering is also used to store partial copy of the central database, specific to particular medical center.

4. Java Integration Platform

Use of distributed architecture in medical system brings abovementioned advantages, but it also introduces extra level of complexity in terms of necessity of performing data replication between disparate remote databases and central database of medical system.

This issue is especially interesting for the reasons:

- high data volume to be replicated,
- high level of system complexity,
- disparate types of exploited databases,
- physical distance of system component locations.

Both high data volume required to be replicated, and high level of system complexity eliminate purposefulness of creation of proprietary application code dedicated to perform such task. There is number of fundamental reasons for this, among which the most important seems to be, that business logic tightly coupled with application code is difficult and expensive to maintain, and the fact, that it is difficult to create an efficient code combining both technical and business aspects, as this would require dedicated skills and knowledge of specific optimization techniques, which are by themselves completely different subject.

Presented system specificity also excludes utilization of solutions specific for particular vendor of database, as at this moment there are 3 implementations in use, namely Oracle, MySql and Access, additionally next database products are foreseen to be used together with further deployments of the system to the successive medical centers.

The next not trivial problem arrives with significant physical distance of system components, combined with low quality computer network between subsystems, which introduces substantial latency and instability of solution based on direct integration, assuming simultaneous reading of data from source and writing to destination database.

Considering depicted issues the decision has been taken to choose alternative approach to component integration, much better fitted to complicated system structure and high data volume. Medical system uses Java Integration Platform, dedicated solution aimed to tackle generic problem of system integration in the data layer, as central service for data replication.

Java Integration Platform is an ETL class system, built using technology of distributed programming components. JIP is entirely coded using Java language, and it uses many extensions of J2EE technology, which gives it solid backgrounds of platform based on open standards, and at the same time enabling reaching stable and efficient solution. The system provides user with easy access to configuration parameters having fundamental impact on efficiency of replication, and it realizes full separation of configuration of particular data interface from integration kernel, giving possibility of storing entire business logic in a form of configuration files in XML format. Providing access to disparate databases, MS Excel files, and any form of text format is built directly into system kernel, equipping user with flexibility of changing data source or target system with minimal amount of changes to interface configuration. In typical cases the only difference is JDBC driver to be used and URL specific to particular connection.

Problem of integration in case of significant physical distance of system components or in case of poor network quality is solved via introduction of indirect replication (Fig. 3). In this scenario they are used 2 cooperating with each other instances of JIP platform. One of them operates on the site of source system and performs local replication from source database to compressed text format and sends such prepared data chunks via network using speed efficient FTP protocol. Another replication domain is deployed locally at the target system site. It receives incoming data, decompresses it, and loads into the destination database.

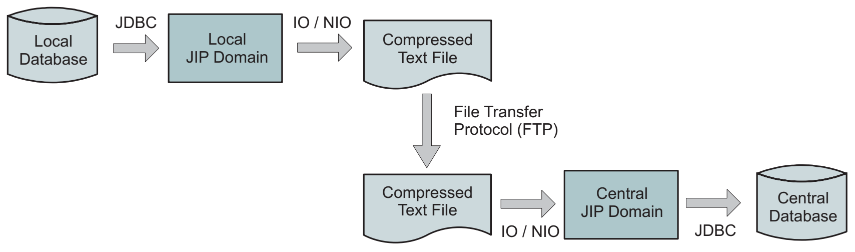

Fig. 3. Indirect integration using FTP protocol

It is worth to mention, that one of the advantages of system architecture using integration platform is full transparency of technical details of data transfer method from point of view of medical system, which acts here as a client expecting data in particular format, but not being interested and aware itself of all potential stages of processing happening between data source and the final data consumer. Such architecture enables dynamic changes to replication structure, without necessity of any interference with medical system.

5. Local databases

Each ward of the hospital, to which discussed medical system is deployed, is equipped with independent local server, communicating with module of central database, and used as a buffer of data collected in real time. Whole service and configuration of connection with the central database is provided in form of a Java Integration Platform domain. At the same time functional part of data queue collected from medical devices is controlled by programmatic drivers dedicated to capture data from these devices.

Medical system captures data from pulse oximeters NPB-295, and respirators CUB 750 as well as Stephania. Pulse oximeter continuously and noninvasively measures functional oxygen saturation of arterial hemoglobin (SpO2) and pulse rate per minute (PR). Respirators in term, besides of their primary role in supplying oxygen or a mixture of oxygen and carbon dioxide for breathing, are used to obtain values of respiration rate (RR), mean arterial pressure (MAP), peak inspiratory pressure (PIP), and positive end-expiratory pressure (PEEP). The devices are connected using serial ports RS-232 to personal computer, on which the local database operates (Fig. 4). System architecture foresees possibility of extension of structure with additional computers connected to the local database.

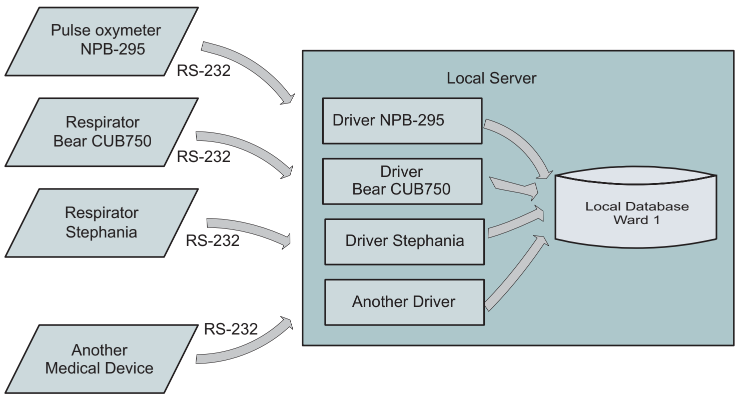

Fig. 4. Local logical system architecture

Data prepared by programmatic drivers are stored in data queue implemented using local database. From there data is further picked up by integration platform and replicated to the central module of the system. Data from respirators are stored every minute, implying creation of over 43 thousands of records monthly, which is not the number causing any performance problems. Situation looks different for pulse oximeters, in which case data is captured every 2 seconds, and it is collected from couple of active sites. In this case monthly data growth is estimated to 6 millions of records, which is equivalent to over 200 MB of data. This is the data volume requiring use of optimization methods in order to achieve satisfying processing time.

There are many factors having significant influence on speed of processing, but it can be approximately assumed, that one month of data from pulse oximeter can be processed on personal computer using integration platform in less than 3 minutes time, which shows, that this mechanism guarantees proper performance even at the time, when medical system is deployed to many successive medical centers.

6. Historical data

One of the common challenges whilst deploying of medical system to the consecutive wards of hospitals is existence of legacy medical systems, based on monolithic, proprietary architecture, storing data in format which is difficult to extract. Such problems came across during extraction of data from Neonatal Information System (NIS). Two phased approach has been used in successfully performed processing. During the first step dedicated solution has been created to connect to NIS database, then integration platform JIP has been applied in order to transform the data and load it into the central database. It is important to stress, that such an approach was necessary, due to complete lack of feasibility of obtaining any standard connection to NIS database. The role of the first stage of processing can be described as implementation of a programmatic driver dedicated to connection to database storing historical data. Second, already standardized step of integration employs kernel of integration platform configured to serve needs of processed medical data. Big advantage of such an approach lies in, similarly as in case of implementation of distributed system architecture, in dynamic description of system configuration stored in files of XML format, giving possibility of easy amendments of configuration following changes to business logic. It is worth of mentioning, that in situation of high system complexity, with necessity of storing data in over 200 tables in the central database of medical system, creation of configuration files itself can be treated as a separate subproject. Presented approach gives possibility of parallelizing of works on the system, and it introduces strict separation between technical and logical system layers.

7. Field of application of integration platform

So far it has been discussed use of integration platform as data replication service in field of data processing between local databases and the central database of medical system and to replicate historical data. The full field of application of the integration platform JIP is completed with:

- data transfer from central server to prediction module,
- replication of predicted medical parameter back to central server,
- data exposition to the user community.

Following the adopted architecture (Fig. 2), module of prediction of medical parameters of patient is deployed to the separate server, instead of being shared with central database server. Such structure causes necessity to replicate input data to the prediction module, and predicted results back to the central database.

The last, but also important way of integration platform utilization is data exposition to the user community. Depending on the needs of particular group of recipients, it is preferred data exposition in a way which is easy for further electronic processing, in which case there are typically delivered text data files, or to the manual analysis, when more natural form is data spreadsheet form or hierarchical XML format. Application of integration module fully covers functional needs of medical system, enabling easy configuration and making of adjustments, together with possibility of fast delivery of data in many formats simultaneously.

8. Conclusions

Introduction of Java Integration Platform as central data replication service in the distributed medical system contributed to significant increase of system flexibility, considered as ease of system adoption to changeable needs. At the same time it has been ensured possibility of extension of the system with new wards of hospitals in non intrusive way, based merely on configuration changes to the data replication process. Good performance has been reached along with high stability of solution. Additionally it has been gained positive side effect in form of creation of data backup in format independent from database vendor.

It is also necessary to point, that application of data integration platform, despite of its many advantages, did not solve all of the problems occurring during system creation. There is still need for creation of dedicated programmatic drivers to communicate with particular types of medical devices. Also connections to non standard databases require creation of dedicated drivers enabling data extraction.

References

[1] Tanenbaum A., van Steen M.: *Systemy rozproszone. Zasady i paradygmaty.* Wydawnictwa Naukowo-Techniczne 2006

[2] Garcia-Molina H., Ullman D., Widom J.: *Systemy baz danych. Pełny wykład.* Wydawnictwa Naukowo-Techniczne 2006

[3] Gold-Bernstein B., Ruh W.: *Enterprise Integration. The Essential Guide to Integration Solutions.* Addison-Wesley Professional 2004

[4] Hohpe G., Woolf B.: *Enterprise Integration Patterns. Designing, Building, and Deploying Messaging Solutions.* Addison-Wesley Professional 2003

[5] Prencipe A., Davies A., Hobday M.: *The Business of Systems Integration.* Oxford University Press; New Edition 2005

[6] Caserta J., Kimball R.: *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleanin.* John Wiley & Sons 2004