Linking knowledge based functions to a commercial PDMS: achievements, obstacles and limitations

Stefan Kraus, Ixchel Castellanos, Hans-Ulrich Prokosch, Thomas Bu¨ rkle

University of Erlangen, Erlangen, Germany

Introduction

Erlangen University Hospital is a tertiary care university hospital with
1,400 beds. A hospital information system with clinical workstations has been in use for several years (1). Since 2006, a commercial patient data management system (PDMS) has been rolled out successively to include 8 intensive care units of the hospital, covering about 100 beds of surgical, neurosurgical, medical and paediatric intensive care (2). At the time of introduction, the selected PDMS covered neither knowledge based functions (KBF) nor advanced calculations e.g. for paediatric scores. Clinicians asked for system enhancement, thus promoting the formation of a funded development cooperation with the PDMS vendor to integrate the PDMS with a rules engine. The Arden Syntax standard (3) was selected to develop such KBF and more advanced calculation schemes.

The goals of the development project described here were

- To find out if a commercial PDMS acting mostly as a black box system with a set of import and export mechanisms could be fitted with a rules engine
- To define a compact and easy to maintain interface between the
PDMS and the rules engine
- To describe the architecture and the limitations such a solution would have

Background

The Arden syntax and its Medical Logic Modules (MLMs) have been extensively described as a potential mechanism for standardized knowledge representation and knowledge transfer between institu- tions (4–6). The requirements for integrating Arden rules into clinical information systems have been repeatedly discussed (7, 8), especially the required Arden compiler (9, 10) and inference engine with trigger mechanisms for evoking MLMs (11). Some commercial clinical information systems such as Agfa Orbis or Siemens Soarian incor- porate an Arden compiler and an inference engine. Integration with a more or less black box type commercial clinical information system has rarely been described (12, 13).

Methods

- A stepwise engineering approach was used to start the integration efforts. Five steps were necessary to perform the integration.
- The first step comprised analysis of the interfaces of the PDMS and the Arden engine. After gaining an understanding of the existing interfaces they were examined for possible ways of linking them
- The second step covered the implementation of a connection between the components. This required two major workarounds on the PDMS side. The main task was to provide the simplest method for mapping patient data onto Arden Syntax data types.
- The third step comprised the development of a user interface to
display the results of the rules engine. Therefore a simple HTML viewer was connected to the PDMS and different communication endpoints were implemented in the host interface of the rules engine.
- The fourth step involved the creation of effective mechanisms for refining the generated information. Physicians have been inter- viewed about the most essential features. These were implemented successively within Arden Syntax submodules.
- The fifth step concerned the suitability for more than one ICU with different setups in a multi client capability environment, which required additional measures to filter events and rules for each client.

Results

It proved possible to connect the PDMS and the Arden rules engine and make KBF and advanced score calculations available for day-to- day work. Today a total of 22 MLMs covering the fields calculation formulae (e.g. calculation of anion gap), margin checking (e.g. checking sodium and potassium values), data-triggered monitoring (e.g. low glucose warning), complex score calculation (e.g. PRISM III scoring) and timeline profiles (e.g. Murray score profile over time) have been implemented for use in routine work.

Analysis results did not show major obstacles regarding the rules engine. It has a generic interface that allows easy mapping of patient data to Arden syntax data types. Furthermore, it supports arbitrary events which can be transmitted to the rules engine and is able to send different types of messages to various endpoints. The interface of the PDMS, however, was not immediately suitable for a simple connection. Being a black box, the PDMS offered an interface to periodically export patient data into text files. Originally designed for printing textual reports and doctor's letters, the interface was later enhanced e.g. to support the creation of user defined html reports in a web interface. The export interface could be invoked user-driven by button or alternatively time-

driven with a cron job. There was neither information about data types nor mechanisms for a data-driven export. Invoking the export from outside the PDMS to access patient data by an external component did not prove possible.
The rules engine was provided by the vendor as a pure library, aninterface for reading and writing data was implemented by ourselves. We integrated the engine into a standalone java application server and developed a web service interface that provides only one method (''report_event'') in production mode. This method requires at least the name of the event and the patient's PDMS-internal case number. The rules engine was enabled to access SQL databases and web services and send emails as well as SMS messages to DECT phones.
As there is no external access to PDMS patient data available, required patient data are periodically exported into text files. The text files are then parsed and the patient data are written into an external database. This database named ''proxyDB'' is a proxy for the PDMS internal database. To enable the mapping of patient data to the proxyDB, a simple, easy-to -parse tagged data format comprising lists of value/timestamp pairs was created.
Because no trigger mechanism was originally provided with the PDMS, a rather basic mechanism was developed to continuously compare each export with previously exported values. If a value is exported for the first time, a data-driven event is reported to the inference engine. In addition, a fixed set of common required vari- ables such as patient name and date of birth is supplied with every event message.
An in-house developed plain HTML-viewer was connected to the PDMS to display the results of user-driven MLMs, which generate HTML and JavaScript output. To avoid the extra effort for manual generation of HTML-code, a set of easy-to-use sub-MLMs was implemented to provide text formatting and table representation, allow displaying and hiding text blocks and tables and create scalable line plots.
The architecture became more complex when the system was expanded to further ICUs. The PDMS maintains all patient data centrally with attributes determining which client ICU the patient belongs to. A PDMS client of any one ICU, however, cannot export data for patients of another ''foreign'' ICU. Therefore it was necessary to implement separate export functions for each ICU. This had the positive side effect of a scalable distributed architecture with the ability to maintain load balancing and adequate performance even for complex KBF.

Discussion
Although integration was successful, it is based to a certain extent on workarounds and some drawbacks remain. Central problems during the process of integration were the lack of essential interfaces for data access and data-driven triggering on the PDMS side. The existing PDMS export mechanism presents a defined interface which did not necessitate any PDMS reprogramming. We found it sufficiently fast and able to retrieve all required patient data items. Surprisingly, it was possible to develop a copying process into the proxyDB and a data comparison technology which was still fast enough for many of the implemented KBF in use. Nevertheless, the time delay in event detection and the additional workload when constantly exporting large amounts of replicate data to the proxyDB is a disadvantage of our solution and the existing PDMS export mechanism has somewhat limited capabilities for this use case. For a multi-client environment, one PDMS client per ICU is required solely for the task of data export, replication and event detection. Potentially, the temporal delay could be minimized by separating the exports for data repli- cation from the exports for detecting events. In this case, however, there is a risk of temporarily inconsistent data in the proxyDB.
A better alternative would be event notification on the side of the PDMS e.g. via HL7-outbound interface, potentially combined with patient data access on service level or API basis. The need for data replication would then be avoided. One technique that is widely used when providing KBF functionality for clinical information systems is to duplicate the stream of HL7 messages at the communication engine. We did this in the case of microbiology data, which could not be retrieved in sufficient granularity for KBF from the PDMS itself. Potentially, this could reduce the delay in event detection for those events triggered by data flowing through the communication engine, but it does not ensure that data from other sources is replicated in time into the proxyDB. Thus, it would still be a workaround, because different data sources are not synchronized, i.e. some data, although already contained in the PDMS database, may not be accessed by MLMs at trigger time.
The lack of suitable interfaces on the PDMS' side is clearly the main obstacle to progressing beyond the state of an advanced pro- totype. The general need for suitable and well defined interfaces to enable easy integration of a rules engine or other external components in clinical information systems such as PDMS becomes obvious again. This is also a legal issue in Europe, because enabling KBF may change the status of a PDMS from a documentation system into a medical device. Many vendors want to avoid this because of the certification requirement.

Future Prospects
Our project is still at the prototype stage and some optimization must be carried out to bring it to an advanced level. This includes a secure messaging mechanism, i.e. acknowledgement of SMS mes- sages sent to DECT phones as well as a PDMS internal warning mechanism, setting icons and displaying messages on the GUI of the PDMS and re-importing generated results into the PDMS. Another important task is to further simplify the

communication and presentation mechanisms in order to encourage physicians and care- takers to contribute as knowledge engineers, specifying their require- ments to KBF.

References

1. Haux R, Seggewies C, Baldauf-Sobez W, Kullmann P, Reichert H, Luedecke L, et al. Soarian–workflow management applied for health care. Methods Inf Med. 2003;42(1):25–36.
2. Burkle T, Castellanos I, Tech H, Prokosch HU. Implementation of a patient data management system—an evaluation study of workflow alterations. Stud Health Technol Inform.160(Pt 2): 1256–60.
3. V2.5-2005 AHA. Health Level Seven Arden Syntax, Version 2.5 (revision of ANSI/HL7 Arden V2.1-2002). 2005.
4. Pryor TA, Hripcsak G. The Arden syntax for medical logic modules. Int J Clin Monit Comput. 1993 Nov;10(4):215–24.
5. Hripcsak G, Ludemann P, Pryor TA, Wigertz OB, Clayton PD. Rationale for the Arden Syntax. Comput Biomed Res. 1994 Aug;27(4):291–324.
6. Pryor TA, Hripcsak G. Sharing MLM's: an experiment between Columbia-Presbyterian and LDS Hospital. Proc Annu Symp Comput Appl Med Care. 1993:399–403.
7. Tafazzoli AG, Altmann U, Wachter W, Katz FR, Holzer S, Dudeck J. Integrated knowledge-based functions in a hospital cancer registry–specific requirements for routine applicability. Proc Amia Symp. 1999:410–4.
8. Hripcsak G, Johnson SB, Clayton PD. Desperately seeking data: knowledge base-database links. Proc Annu Symp Comput Appl Med Care. 1993:639–43.
9. Karadimas HC, Chailloleau C, Hemery F, Simonnet J, Lepage E. Arden/J: an architecture for MLM execution on the Java platform. J Am Med Inform Assoc. 2002 Jul-Aug;9(4):359–68.
10. Kuhn RA, Reider RS. A C?? framework for developing Medical Logic Modules and an Arden Syntax compiler. Comput Biol Med. 1994 Sep;24(5):365–70.
11. Hripcsak G, Clayton PD, Jenders RA, Cimino JJ, Johnson SB. Design of a clinical event monitor. Comput Biomed Res. 1996 Jun;29(3):194–221.
12. Burkle T, Prokosch HU, Hussak G, Dudeck J. Knowledge based functions for routine use at a German university hospital setting: the issue of fine tuning. Proc AMIA Annu Fall Symp. 1997: 61–5.
13. Cho I, Kim J, Kim JH, Kim HY, Kim Y. Design and implementation of a standards-based interoperable clinical decision support architecture in the context of the Korean EHR. Int J Med Inform. Sep;79(9):611–22