Workflow Technology to Enrich a Computerized Clinical Chart with Decision Support Facilities

Silvia Panzarasa¹, MS, Silvana Quaglini, PhD², PhD, Anna Cavallini³, MD, Giuseppe Micieli³, MD, Corrado Pernice⁴, Mauro Pessina⁴, Mario Stefanelli², PhD

¹CBIM Consorzio di Bioingegneria e Informatica Medica- Pavia, Italy
²Department of Computer Science and Systems, University of Pavia, Italy
³UC Mal. Cerebrovascolari/Stroke Unit, IRCCS C. Mondino Foundation, Pavia, Italy
⁴TSD-projects, Milan, Italy

silvia.panzarasa@unipv.it

Abstract

Literature results and personal experience show that intrusive modalities of presenting suggestions of computerized clinical practice guidelines are detrimental to the routine use of an information system. This paper describes a solution for smoothly integrating a guideline-based decision support system into an existing computerized clinical chart for patients admitted to a Stroke Unit. Since many years, the healthcare personnel were using a commercial product for the ordinary patients’ data management, and they were satisfied with it. Thus, the decision support system has been integrated keeping attention to minimize changes and preserve existing human-computer interaction. Our decision support system is based on workflow technology. The paper illustrates the middleware layer developed to allow communication between the workflow management system and the clinical chart. At the same time, the consequent modification of the graphical users’ interface is illustrated.

Introduction

It is well known that paper-based clinical guidelines (GLs) are difficult to be adequately distributed and used by final users. The medical informatics community hypothesised that more formal electronic versions would increase physicians compliance to GLs. This claim was based on some hypotheses: one is that formal models, such as flowcharts or other graphical metaphors, provide a more friendly and immediate way of reading a GL, thus improving learning and interiorization of the GL content (e.g. using formal models as educational/simulation tools); another hypothesis is that formal models allow easier GL integration with electronic patient record (EPR), raising possibility of building real-time decision support systems, that again should improve physicians’ compliance.

Concerning the first hypothesis, it is not so straightforward that physicians prefer to navigate a flowchart instead of reading a booklet. At least the flowchart must be enriched with pointers to text and literature, and users must be trained to the new formalism. Concerning the second hypothesis, some evidence exists in literature¹,², but also contradictory results are reported³,⁴.

In fact, while in the past years the emphasis has been put on how to formally represent a GL, now the big issue in this field is how to best integrate it in real healthcare settings, taking into consideration the current local work organization and the different needs that users perceive to tackle different problems of the every-day practice. The following is a non-exhaustive list of questions to be answered before such a system is implemented:

- Does the GL have to be explicitly shown with navigation facilities?
- If yes, does it have to be shown entirely or partially, e.g. according to patient’s clinical status?
- If yes, when does it have to be shown? On user’s request or automatically, according to some trigger? And how (e.g. as a flowchart or a piece of text)?
- If no, the GL engine will run behind the system: how to show the inferred recommendation? As a message in a pop-up window (synchronously), as a short message service (sms), or as a less intrusive communication that may be read (asynchronously) when the user decides to read it?
- If a non compliance is detected, does the user have to provide a motivation? If yes, when? Immediately or when he decides to do it, or during a pre-defined event, for example the daily meeting (if any) or the patient discharge?

Answering these questions is particularly challenging when an information system already exists in the clinical environment, and users are
familiar with a given interface. In this case, it is very difficult for them to shift to other systems or to use parallel ones. Medical informatics and artificial intelligence experienced several failures in the past, due to this problem. Learning new interfaces, double data input, boring interaction modalities have been the main causes of early abandon of several information systems and expert systems in medicine.

To overcome these problems, while developing the application described in this work, that concerns the integration of a GL for stroke patients into an existing information system, the driving idea has been that users must perceive the new system as an upgrade of the previous one, just providing some additional functionalities to the usual interface. All steps of the new system design and development have been performed in collaboration with the technical team responsible for the EPR and the healthcare personnel of the involved stroke units. We stress the paramount importance of this knowledge and experience sharing among working teams in order to give the final tool a high probability of success and users’ satisfaction.

### The former information system

A project funded by the Italian Ministry of Health, required to provide about 20 Neurological Departments with a decision support system, based on “The Italian guidelines for stroke prevention and management”, delivered by the SPREAD association. Some hospitals were already equipped with a Computerized Clinical Chart (CCC) implemented, some years ago, with the commercial tool Wincare® (by TSD Projects). User’s interaction with this tool is based on modules: a module is a single form of the clinical chart, through which it is possible collecting patient data and generating textual reports. In fact, to each module a data entering form is associated. To access the system, a username and a password are required, grouped on the basis of the role (physician, professional nurse, physiotherapist, etc) in order to allow the visualization of competent modules only (see Fig. 1). As shown, the list of the modules is not structured at all: it is given in alphabetic order and it is not tailored to the specific patient.

Another characteristic of this CCC was the presence of several free text data entry fields. On the foreground, different module lists shown to physicians (left) and to nurses (right) personnel with the new decision support functionalities. Since users, as already mentioned, were satisfied with their CCC, as the first step we analyzed how many changes were necessary in order to support the GL integration, with the aim of minimizing new interaction modalities. To make an example of a necessary compromise we achieved between users’ needs and GL engine needs, let us consider the patient’s and family’s histories: they were traditionally entered as free text. This was not compatible with the need of interpreting rules such as: “IF the patient had a previous myocardial...”

### Methodology for the GL integration

To add the decision-support functionalities, which, among other improvements, will create dynamic list of modules, reengineering of both structure and interface of the EPR was performed. A deep analysis of the SPREAD GL, the existing EPR, and its interface, allowed to determine the minimum data set required for implementing all the GL recommendations, to check whether these data were or not already managed by Wincare®, and whether they were in the opportune format (e.g. encoded versus free text). Moreover, the analysis of the clinical workflow in the Stroke Units allowed devising an adequate interaction modality of the...
infarction, THEN prescribe ASA drugs”. Previous myocardial infarction of course must be encoded. Thus, a complete restyling has been necessary, and all the clinical history information has been encoded through ICD9-CM. By sure, encoding has been an advantage for the GL engine implementation, but also a disadvantage for another aspect of the clinical workflow: in fact, physicians were used to print out the clinical histories and attach them to the discharge letter, “as they were”, since they were stored in natural language. As a consequence, to make the ICD9-CM encoding acceptable to final users, a parallel tool has been developed, which translates the encoded diseases, plus some associated comments, into natural language sentences.

Implementation
After the data model analysis, and its consequent update, we implemented the GL recommendations through a Workflow Model, using Oracle Workflow™, thus developing a so-called Careflow Management System™ (CfMS). It manages the execution of GLs through a workflow engine, interpreting the care process definition and interacting with users (called Cf participants). An instance of the Cf is the execution of the care process for a particular patient, specifying which medical tasks need to be executed, in which order, and by which professional role.

The Cf model (see a portion in Fig. 2), is described on the basis of the SPREAD GL, with some site-specifications decided by the involved neurologists. All the GL recommendations were implemented, regardless of their scientific evidence level (SPREAD uses four levels, from A to D, where A is the highest one and stands for scientific evidence coming from meta-analysis or clinical trials, while D is the lowest one and stands for knowledge derived from case reports or expert consensus).

The CfMS needs patient data in order to feed the workflow engine and to interpret GL rules, then it must be able to communicate the patient-specific recommendations to the users, or Cf participants, through messages and to-do-lists. As already said, our choice was to integrate all the needed functionalities within the existing end-user application, making it more “dynamic”, according to the CfMS execution, without creating a new specific interface.

Technical functionalities
The middleware layer shown in Fig. 3 has been developed to keep the systems independent, while granting communication. It is composed by a supporting database (Oracle DB) and an Interpreter written in PL/SQL.

The middleware DB contains a set of tables with static content and a set of dynamic tables which content is patient-specific. The most significant tables in these sets are: Transcoding-Task; Transcoding-Data and Mapping (static); Active-Patients; Messages, To-do-list, and Data-exchange (dynamic).

The interaction is as follows: once a patient is admitted to the Stroke Unit, the first Wincare® module called by a physician or a nurse is Admission: on every new admission, immediately a new instance of the CfMS is activated by Wincare®, that inserts the patient ID into the table Active-Patients. Then, whenever there is a new data entry in Wincare® for a patient whose ID is in Active-Patients, the attributes that are useful for the GL interpretation are transferred into the table Data-exchange, in accordance to the information stored in Transcoding-Data (see Fig. 4). Transcoding-Data contains, for each Wincare® MODULE, the

<table>
<thead>
<tr>
<th>P_ID</th>
<th>EVENT</th>
<th>DATA</th>
<th>DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>OBJECTIVE_EXAM</td>
<td>80/1.67</td>
<td>02/15/2006</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EVENT</th>
<th>DATAFLOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJECTIVE_EXAM</td>
<td>WEIGHT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EVENT</th>
<th>TABLE_NAME</th>
<th>FIELDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJECTIVE_EXAM</td>
<td>AGE</td>
<td>DOB</td>
</tr>
</tbody>
</table>

Fig. 4 – Attribute values in some rows of the middleware DB tables

DATAFLOW, i.e. the sequence of the required attributes, with names separated by the special character “§”. These attributes are a subset of the complete form, and belong to the minimum data set mentioned above. We remark that every future change of the required attributes or modules, for example due to periodic revisions of the SPREAD
GL, will only imply an update of the static tables in the middleware DB. As a new record is inserted into Data-exchange, a trigger activates the Interpreter: its function is to read Data-exchange and to create a dynamic SQL query to store patient data in the right CfMS table, thanks to Mapping, which contains the name of this table and the corresponding sequence of fields. According to the data shown in Fig. 4 the query generated by the Interpreter will be “insert into STAGE.OBJ_EXAM (P_ID, DATE, P_WEIGHT, P_HEIGHT, P_BMI) VALUES (‘1_75’, ‘02/15/2006’, ‘80’, ‘1.65’, ‘29.4’);”.

On the other side, when the CfMS generates a suggestion, or a new list of Modules, they are put into the tables Messages and To-do-list of the middleware DB, and Wincare® can read and show them through its interface. To refresh the user's interface, Wincare® executes a reading of these tables every minute. The same tables can also be updated by Wincare® and then read by the CfMS, again via the middleware DB. This may happen either when a user answers to a communication (table Messages), or when a user modifies the to-do list by adding modules different from those proposed by the CfMS (table To-do-list).

The Messages table is composed by the following fields: P_ID (patient id), M_ID (a unique id for the message), MESSAGE (the text of the message or the suggestion), RECEIVER (target role of the message), PRIORITY (it modulates the message visualization and it can be low, medium, or high), MODALITY (if the message is read-only or if it requires a response), RESPONSE (the response of the receiver to the message, only yes or no).

The To-do-list table contains the patient-specific list of modules: it is very “dynamic” because it changes according to patient-data and response to the care process. In addition to the fields P_ID and Module, there are POSITION, that specifies the order of execution, and STATUS, that specifies the status of the module (see next paragraph).

The new version of the user’s interface

Fig. 5 illustrates the new list of modules: with respect to the former system (Fig. 1), where the list was unstructured and static, now it is structured and dynamic, because it is built on the patient's data and CfMS suggestions via the To-do-list table. Date and time of last update are shown on top of the list. Different icons and colors are used to represent the field STATUS, whose values are:

- NULL- the module has still to be done (no icon);
- OK- the module is complete (green happy smiley);
- EXEC- the module is currently being done (yellow thoughtful smiley);
- EXC- the module has not been executed due to an exception (red sad smiley);
- WINCARE- the module has been added by the users (Wincare® icon).

Together with the STATUS, also the grade of recommendation is shown, as in the SPREAD GL, by letters A-D.

According to the physician needs, special attention has been put in avoiding over-information: particularly in acute situations, only modules necessary to face the urgency are listed.

An example of the importance to keep the list of modules updated is the management of thrombolysis for a patient admitted to the Stroke Unit within less than three hours from the stroke onset (therapeutic window). In Fig. 5a, the patient is potentially eligible for the thrombolytic drug r-tPA, according to information coming from the recent history module. Since this treatment is both extremely important (SPREAD level of recommendation A) and delicate (it may cause hemorrhages), the task list shows only the mandatory modules to be considered in order to detect possible contraindications. To facilitate users, the module “Check for r.tPA Eligibility”, which in Fig. 5 is in the EXEC status, reports the complete list of contraindications: this list is automatically filled by the CfMS, that continuously reads data from the EPR, but also it may be filled manually by the

![Fig. 5 – Lists of modules for the same patient in two different points of his/her care process](image-url)
physician if the situation is very urgent. For example, CT scan could provide some findings advising against r-tPA. Fig. 6 shows the corresponding Wincare® module’s interface: the findings relevant for the GL, which will be transferred to the middleware DB, are marked in yellow, in order to encourage a correct and complete editing of the EPR.

Let us imagine that TC scan shows an intracranial haemorrhage: in this case, r-tPA cannot be administered. Consequently, as shown in Fig. 5b, “Check for r-tPA eligibility” and “r-tPA Treatment” switch to the exception status (red sad smiley).

Fig. 6 – The user interface for the module CT scan

Contemporary, also less urgent modules appear in the to-do list. Double clicking on the red modules allows knowing the reasons for this exception.

To avoid intrusiveness, when the CIMS suggestions are neither urgent nor critical, they are translated into simple communications, stored in table Messages and sent to the appropriate roles. These messages are visualized in a separate window, the communication box, accessible by clicking an icon in the right bottom of the screen, which becomes red only when a new message arrives. CIMS suggestions are listed grouped by the patient identifier and ordered by date and level of priority (represented by different icons). When an answer is required, a checkbox appears near the text of the message allowing the RECEIVER to say whether he agrees or not with the SPREAD GL.

In the case of positive answer, some data may be stored automatically into the EPR. For example, if a physician accepts a suggestion about a drug prescription, a record with the drug name, dosage, and timing is stored and at the same time a message is sent to the role “nurse”, which is responsible for the drug administration.

Conclusion

We have illustrated the re-engineering of an information system, necessary to improve the system with decision support, primarily based on a user needs analysis. The application described in this paper is a demonstration of how a decision support system may be integrated within a computerized clinical chart without introducing dramatic changes in the end-user application. While the specific application described has been done on top of a specific, commercial, EPR management system, we want to highlight that the middleware layer is general enough to allow integration of different models, implemented through CIMSs, with different EPR management systems. The actual system is working for ischemic stroke patients only: after collecting some additional users’ feedback, the system will be adapted to the management of all stroke patients.

References


Acknowledgements

This work has been partially funded by projects FIRB (Italian Ministry of University and Research) and STAGE (Italian Ministry of Health). We thank Stefano Solari (TSD Projects) for helpful discussion on technical implementation aspects.