

A Guideline Management System

Paolo Ciccarese^a, Ezio Caffi^b, Lorenzo Boiocchi^a, Silvana Quaglini^a, Mario Stefanelli^a

^aDipartimento di Informatica e Sistemistica, Università di Pavia, Italy
^bConsorzio di Bioingegneria e Informatica Medica, Pavia, Italy

Abstract

This paper describes the architecture of NewGuide, a guideline management system for handling the whole life cycle of a computerized clinical practice guideline. NewGuide components are organized in a distributed architecture: an editor to formalize guidelines, a repository to store them, an inference engine to implement guidelines instances in a multi-user environment, and a reporting system storing the guidelines logs in order to be able to completely trace any individual physician guideline-based decision process. There is a system “central level” that maintains official versions of the guidelines, and local Healthcare Organizations may download and implement them according to their needs. The architecture has been implemented using the Java 2 Enterprise Edition (J2EE) platform. Simple Object Access Protocol (SOAP) and a set of contracts are the key factors for the integration of NewGuide with healthcare legacy systems. They allow maintaining unchanged legacy user interfaces and connecting the system with whatever electronic patient record. The system functionality will be illustrated in three different contexts: homecare-based pressure ulcer prevention, acute ischemic stroke treatment and heart failure management by general practitioners.

Keywords:

Clinical practice guidelines, guideline management system, careflow management system, heart failure, pressure ulcer, stroke.

Introduction

Clinical practice guidelines (GLs) are a means to improve the quality of delivered health care. There are several official web sites, such as National Guideline Clearinghouse (www.guidelines.gov), publishing GLs with the aim of diffusing them within the medical community. Most of them offer GLs in textual format with some useful hyperlinks. Moreover, tools for computerized GLs representation have been developed in the last decade [1]. Among the well agreed-on benefits of a GL computerisation, with respect to the traditional textual format, there are the text disambiguation, the possibility of viewing the guideline at different levels of details without losing the entire view, and the possibility of generating patient-tailored suggestions. Different papers have been re-

cently published to evaluate the real effect of GLs in the everyday usage [2]. In particular, when systems do not show a clear benefit, this could be due to the poor user’s willingness to use them [3]. In turn, poor willingness could derive from low integration degree with the existing legacy systems and low consideration of organizational aspects. Our architecture reduces the final user effort to learn and use new interfaces, by integrating the three aspects characterizing an innovative Health Information Systems (HIS):

1. patient data stored into an Electronic Patient Record (EPR),
2. medical decision support, through a Guideline Management System (GIMS),
3. organizational support, through Careflow Management System (CfMS).

At the user level, there will be a unique interface managing the three aspects. To reach this integration, we reused knowledge and experience acquired in developing earlier systems [4,5], keeping attention to the separation of concerns (SoC). This issue is illustrated in the architecture shown in fig. 1, where a CfMS is viewed as an external service for GIMS and vice versa. Data and ontologies are shared through the entire

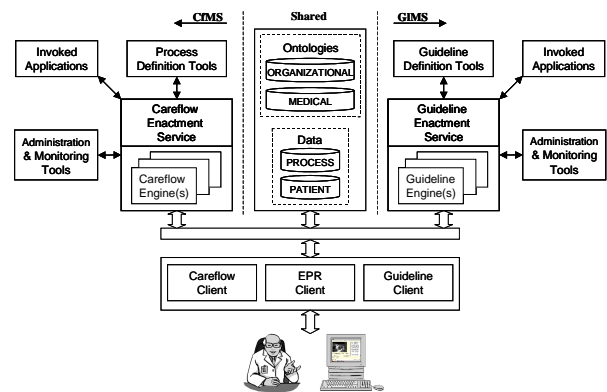


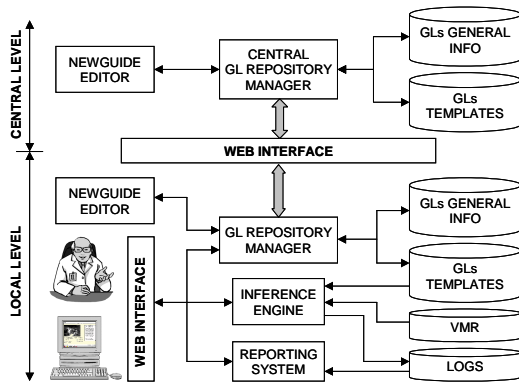
Figure 1- The conceptual system architecture

system. The organizational model is defined and used by the CfMS while the medical knowledge is defined and used by the GIMS. The interaction between such systems is message-based according to specific contracts. A domain doesn’t need to know anything more than the contract details and the shared ontologies to communicate with the other one. The

strong separation between medical and organizational issues allows improving not only the software product (reducing crosstalk and powering parallelization) but also the quality of interaction between the system and its administrators who usually have different skills and background.

The GIMS Architecture

NewGuide is a distributed environment aiming at the complete management of computerized GLs life cycle. It is based on two main levels: a central one and a local one. The central one could be regional, national or international, according to the healthcare delivery policy. At the central level the goal is to represent, distribute and manage GLs, certified by some health authority or scientific organization. At the local level, healthcare organizations (HCOs) may decide to adopt one of such GLs to increase the quality of the care services. Site specifications may be performed by individual HCOs. We strongly believe that also these site specifications have to be made available at a central level also, after a certification process that guarantees the quality of the modified GL. Two advantages may derive: avoiding that different HCOs make the same changes independently, and managing the process of GLs updating (if several users modify a GL it is very likely that the original GL needs to be updated). Also the GL performance evaluation requires exactly knowing which version or site specification of the GL entered into the clinical practice.



In the following, we illustrate how a GL can be formalized, uploaded, downloaded and implemented in real-world settings. As shown in fig. 2, computerized GL lifecycle starts with the NewGuide graphical editor application module, through which the GL can be represented as a set of decision flow-charts that involve conceptual entities, such as medical terms and concepts [6]. Every GL generated by the editor, either at central or local level, is split by the GL repository manager into two databases: the first one stores the general information about the GL (aim, eligibility, authors, version, etc.), and the second one the template, i.e. the decision flow-model together with its related concepts. The HCO can select the appropriate GLs through the general information stored in the repository manager. Ideally the responsibility for choosing and implementing GLs into a HCO should be up to the “health

care processes management” department. It will also define user groups and a hierarchy among them assessing priorities and permissions. The final user invokes the inference engine and creates an instance of the GL for the management of an individual patient. This requires data from a Virtual Medical Record (VMR) [7]. VMR is the NewGuide middle layer that stores every kind of patient information either acquired through a legacy system (HIS) or entered by the GL user. Each inference engine step implies both producing recommendations, such as a drug prescriptions or laboratory tests, and updating a logs database. The latter contains care process information such as progress status of each GL task with relative time stamps. It is worth underlining that the system is multi-user: several physicians can instantiate the same guideline for different patients. Of course, only one physician can control the GL flow on the same patient at the same time.

The following sections will describe in detail all the components of the NewGuide architecture displayed in Fig. 2.

The NewGuide Editor

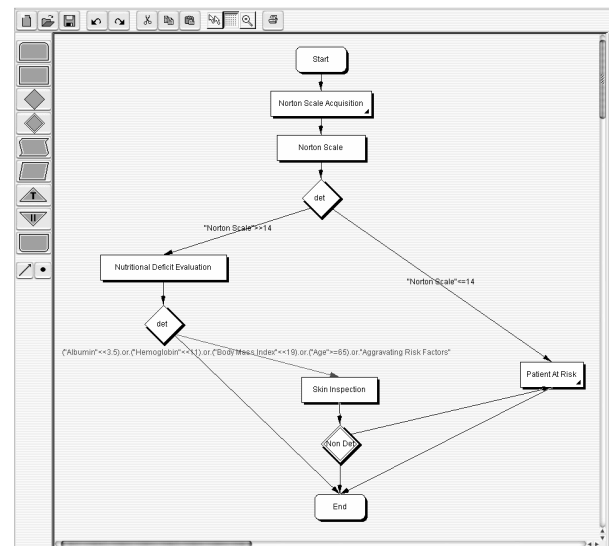


Figure 3- First Level of pressure ulcer prevention GL

The NewGuide editor, as extensively described elsewhere [4,6], is a graphical environment that allows the guideline developer to formally represent clinical GLs and protocols. The representation approach is flow-chart like, as shown in fig. 3. The main components are tasks (rectangles), and decision points (diamonds). Given the complexity of the care delivery processes, we adopt a multi-level representation. Thus, some tasks in fig. 3 can be expanded into a lower, more detailed, level. The NewGuide editor produces four XML data structures containing the following information or knowledge:

1. The GL general properties according to the GEM model [8], such as identity, developer, purpose, intended audience, method of development, target population, testing and revision plan. In particular the purpose and its quantification modalities are important for the GL performance evaluation.

- The set of medical terms (such as clinical observations, diagnoses and therapies) and their associated codes used by the GL. They will be a subset of the standard ICD9-CM and LOINC terminologies [9] plus a set of terms entered by the GL developer since he did not find them in the above mentioned terminology servers.
- The set of abstractions, both qualitative and temporal [10], needed to represent complex concepts, such as: "the patient had abnormally high serum creatinine after taking ace-inhibitors for less than two weeks", where "abnormally high serum creatinine" is a qualitative abstraction indicating "creatinine>2mg/dl", and "ace-inhibitors for less than two weeks" is a temporal abstraction.
- The GL flow, indicating the activities and the decision processes, as represented in the graphical layout.

The NewGuide Repository

The repository module manages the data structures generated by the GL editor. The GL general properties are used by the repository manager itself to assist the user in the selection of the appropriate GL (see fig. 4). The other data structures describing the GL (points 2-4 in previous section) are stored in a compressed format on the central server. The file will be then expanded at local level into a ready-to-use GL template (i.e. a GL model) on the first user request. Additional repository functionalities, accessible through the NewGuide web interface, are updating and deleting GL versions, as well as performing summary statistics on GLs templates usage.

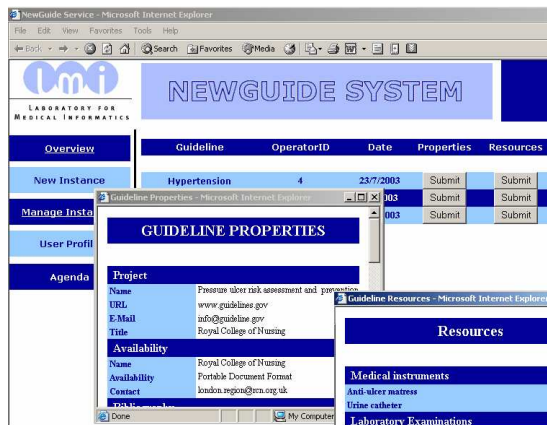


Figure 4- Retrieving the appropriate guideline

Choosing a GL

Among the concepts associated to a GL, the list of all the diagnostic tests, interventions, and therapies that a GL could suggest is particularly useful for potential users. In fact, in addition to the GEM-like general properties, they provide a more informative picture of the GL that allows knowing which data are requested to implement it. Analyzing the list of such data, as shown in fig. 4, the potential user may realize whether his HCO is able to provide all the laboratory and instrumental resources needed, thus having an idea of the implementation difficulties and additional costs (e.g. need for external services).

The Inference Engine

The architecture of the inference engine is shown in fig. 5. As soon as a user asks for a GL from the NewGuide repository, the general manager creates an instance manager, which will enact an instance of that GL (i.e. a GL referring to a particular patient). The instance manager interprets the GL flow and generates suggestions on the basis of the information stored in the VMR. This will allow at any time, together with the logs database, to reconstruct the patient management history. The communication between NewGuide and the external world is managed by the message manager, which delegates requests and responses to the web user interface or to a SOAP interface on the basis of the system configuration.

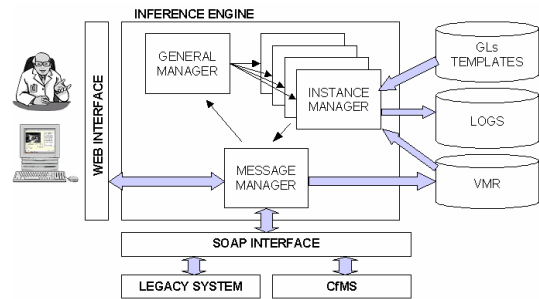


Figure 5- The architecture of the inference engine

SOAP

SOAP [http://www.w3.org/2000/xp/Group/] is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined data types, and a convention for representing remote procedure calls and responses.

GL states

The instance manager is able to start, finish, drop, suspend, resume, stand by, set ready and activate a GL execution. Let us focus on "suspend", "stand by", "ready to be activated" and "activated" states, because they are strongly related to some organizational issues. While "suspend" is a state explicitly requested by the user, and "resume" is the corresponding user request to continue, "stand by" is a state where the system is waiting for external events, such as patient information availability or a user decision. "Stand by" has been introduced to replace a synchronous interaction with an asynchronous one. The responsibility for maintaining the correct GL flow and timing (i.e. the subsequent activation) is left to the external CfMS. We illustrate this functionality with two examples where contracts between GIMS and CIMS are detailed:

- When the GL says "wait for 2 days" it should be inefficient to maintain the GL running in the waiting state, so the GL is put in the "stand by" state and it will be reactivated as soon as the CfMS will set it at "ready to be activated" and the user will activate it. In this way, as previously mentioned, a GL represents medical knowledge only, while a CfMS is responsible for monitoring that user, playing the

suitable role (organizational knowledge), executes the tasks assigned to him. For example, in the case of a recommendation for "waiting for a certain time", the CfMS will alert users that the deadline for a task execution is being achieved.

- When an information acquisition task has to be executed, the inference engine can ask the message manager to send the request through the SOAP interface and the GL is put on "stand-by". As soon as data become available, they are sent back to the message manager, which verifies their validity, stores them in the VMR, and sets the inference engine state at "ready to be activated". The data validity is checked on the basis of the actual GL (i.e. the GL could ask for the last-hour Potassium value), but an agreement could also be achieved with the user who might accept also not valid data if he believes that they are still useful, i.e. validity expired only few minutes before. Of course this will be marked as a non-compliance. In case of non-valid data the CfMS can be committed to force the acquisition of missing information.

The Reporting System

Logs file is filled by the instance manager and contains all the statuses assumed by every GL instance since its creation. Time stamps, current GL step and users identifiers are stored to this aim.

System deployment modalities

In principle, the above-described NewGuide components can be installed on different physical systems: each healthcare setting could have its own installation.

Three different levels of interaction between users and the NewGuide inference engine have been designed and developed in order to leave each HCO to decide which is the most convenient integration of a GL-based decision support service with its working environment (see fig. 6):

- The lowest level does not require any kind of software installation from the user side, since the interaction is completely web-based, with dynamic pages automatically built by the NewGuide web-interface. In this modality, in-

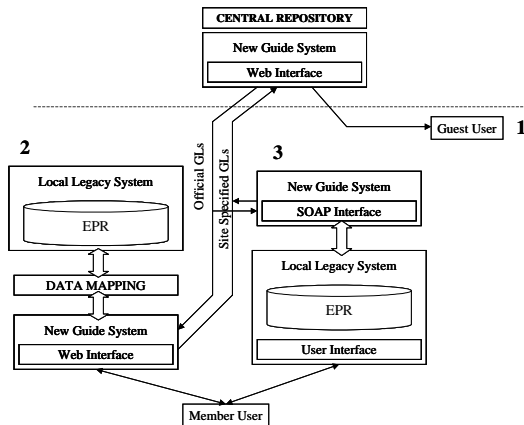


Figure 6- Possible deployment modalities

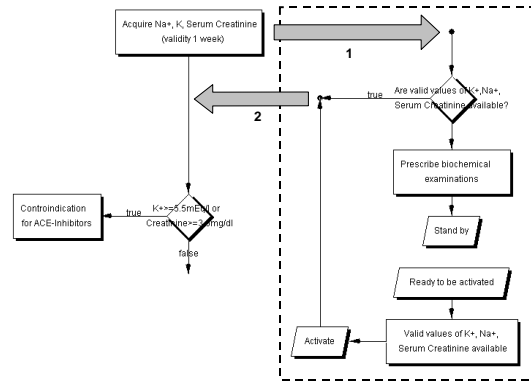


Figure 7- A piece of the GL model and its connected workflow model

put to VMR is completely manual. Such a modality is particularly convenient for educational purposes and when no mapping is possible from the HCO's EPR to the NewGuide VMR.

- The second level of interfacing requires a mapping of the EPR into the NewGuide VMR [6]. The user-system interaction still occurs through the NewGuide web-interface, but patient information is retrieved from the VMR. The user is asked to enter patient information only when it is not available in both the VMR and the EPR.
- The third level of interfacing can be used when NewGuide plays the role of a decision support system under the control of the HIS; this allows the user to interact with NewGuide through the HIS interface. To implement such an interaction modality, we adopt the SOAP standard that requires an agreement on to define a data and command exchange protocol.

Three implementation exemplars

In this section we present three NewGuide services implemented within different HCOs that required using the three different interaction modalities described in the previous section. These applications are paradigmatic because they imply different hardware, users and healthcare settings.

The pressure ulcer prevention GL

This is a homecare application where the operators are nurses assisting elderly or bed-ridden people. For usability reasons (nurses must join patient's home in different geographical regions) the handheld has been adopted as the user's hardware. Through a web-browser using a wireless connection, nurses access the central system, and ask for GL suggestions. Since in this case there could be no EPR and the system may require patient's clinical information, it must be provided by the nurse herself through a direct typing.

The acute ischemic stroke treatment

This is a hospital application, where the users are neurologists taking care of acute patients. Here, there is the need to acquire information from several databases distributed in different

hospitals. Thus we first performed the mapping of these legacy systems to the NewGuide VMR (to this aim, the connectivity commercial product *dbMotion*¹ was used). Suggestions produced by the GL are shown in the NewGuide web-interface style.

The heart failure management

This is an application for the general practitioners (GPs) to be used during encounters with patients at the GP office. Here, the EPR and a CfMS are already available in the healthcare setting as a service provided to the GPs by the regional healthcare system. NewGuide has been required to provide suggestions according to the most recent guidelines for the heart failure management. The necessary clinical data are retrieved via SOAP (without requiring a mapping to the NewGuide VMR) on the basis of an agreement about the medical terminology. The suggestions produced by NewGuide are integrated in the existing user interface in a totally customized form. In this context the stand-by mechanism is particularly useful because the GL holds for a long time and it must manage several encounters. In fig. 7 the left-most portion represents a data acquisition task and a subsequent decision task, formalized through the NewGuide editor. From the point of view of the inference engine, that must fit the real-world interaction between the user and the GL, the two tasks are translated as shown in the dashed portion of the figure. In fact, when not enough recent data are available, the GP prescribes laboratory examinations. At this time, the GL is set in the stand-by state and a message is sent to the CfMS that some data are expected from the laboratory within a certain amount of time. As soon as these data are available, the patient will be advised to come back to the GP for the next encounter. Only at that time the GL will be activated again. It is up to the CfMS to assure that the GL-established deadlines are met.

Conclusions

With the system described in this paper, two objectives are pursued: the first one is to build a GL repository where the knowledge representation is homogeneous (i.e. the same formalism is used to represent and to document every GL). The second one is to provide the medical community with a GL server whose characteristics foster GLs implementation, by allowing the most suitable integration level with the legacy systems. This is the key factor of the NewGuide project: different possibilities in terms of deployment and integration for an effective usage in real practice. Moreover, as a service, we would like also to manage the GLs evaluation, in terms of both outcomes analysis and, more informally, a discussion forum available to all the users and healthcare administrators.

Acknowledgments

This work has been supported in part by a GRANT (RBNE01HWWR_001) from the Italian MIUR (Ministero per l'Istruzione, Università e Ricerca) and in part by the E-Heart project, funded by Provincia di Trento, Italy.

References

- [1] Peleg M, Tu S, Ciccarese P, Kumar A, Quaglini S, Stefanelli M et al. Comparing models of decision and action for guideline-based decision support: a case-study approach. *JAMIA* 2003; vol. 1 ,n.10, 52-68.
- [2] Barner E. S., Baker S, Funkhouser E. et al. Do local opinion leaders augment hospital quality improvement efforts? A randomized trial to promote adherence to unstable angina guidelines. *Med Care* 2003; 41; 420-431.
- [3] Eccles M., McColl E. et al. Effect of computerized evidence based guidelines on management of asthma and angina in adults in primary care: cluster randomised controlled trial. *BMJ* 2002 vol. 325.
- [4] Quaglini S, Dazzi L, Gatti L, Stefanelli M et al. Supporting tools for guideline development and dissemination. *Artificial Intelligence in Medicine* 1998; 14:119-137.
- [5] Panzarasa S, Maddè S, Quaglini S, Pistarini C, Stefanelli M. Evidence-based careflow management systems. *Journal of Biomedical Informatics* 2002; vol. 35, 123-139.
- [6] Ciccarese P., Caffi E., Boiocchi L., Alevi H., Quaglini S., Kumar A., Stefanelli M. The NewGuide Project: guidelines, information sharing and learning from exceptions. *Proceedings AIME* 2003.
- [7] Johnson P, Tu S, Musen MA, Purves I. A Virtual Medical record for Guideline-Based Decision Support. *AMIA Annual Symposium*, Washington, DC, 2001.
- [8] Shiffman RN, Karras BT, Agrawal A, Chen R, Marengo L, Nath S. GEM: A proposal for a more comprehensive guideline document model using XML. *JAMIA* 2000; 7(5): 488-498
- [9] LOINC Users Guide release 1.0N, 02/04/ 2000.
- [10] Bellazzi R, Larizza C, Lanzola G. An http-based server for temporal abstractions. *IDAMAP '99 working notes*, pag. 52 – 62.

Address for correspondence

Paolo Ciccarese, paolo@aim.unipv.it, Dipartimento di Informatica e Sistemistica Università di Pavia, Via Ferrata, 1 - 27100 Pavia, Italy.

¹ DbMotion is a product of Ness-ISI Ltd. Beer Sheva, Israel (www.dbMotion.com)