

GLEE – A Model-Driven Execution System for Computer-Based Implementation of Clinical Practice Guidelines

Dongwen Wang, M.Phil., Edward H. Shortliffe, M.D., Ph.D.
Department of Medical Informatics, Columbia University
New York, NY 10032

We have developed the GLEE system for execution of guidelines encoded in GLIF3 format. This system can be integrated with a local clinical information system through standard interfaces to EMRs and clinical applications. The execution model of GLEE takes the “system suggests and user controls” approach. A tracing system is used to record the state of guideline steps and their transitions. The GLEE system provides an internal event-driven execution model that can be hooked up with a clinical event monitor in local environment. We discussed the execution flexibility provided by GLEE and issues related to its integration with a local clinical information system. Potential use of the GLEE system includes clinical decision support, quality assurance, guideline development and medical education.

INTRODUCTION

In recent years, clinical practice guidelines (CPGs) are developed to address the issue of inappropriate practice variations, with the goal of improvement in care quality and control of costs¹. Computer-based implementation of CPGs integrated with clinical decision support systems have been recommended for this purpose^{1,2}, encouraged by their effectiveness to improve clinical performance and patient outcomes³.

During the last decade, several computer-based models have been developed to represent CPGs in computer-interpretable formats^{4,5}. An important goal to develop these models is to make the encoded guidelines sharable across different institutions to save the huge resource investment in the process of guideline development and implementation. At the same time, the execution flexibility of a guideline system is always a major challenge for guideline application. Finally, difficulty of integration with local electronic medical record (EMR) systems and clinical applications is another issue that needs to be addressed in guideline implementation.

In 1998, the GuideLine Interchange Format (GLIF) was developed by the InterMed Collaboratory as a guideline representation model aimed to share CPGs across different institutions⁶. Later, a prototype

execution engine was designed and implemented to integrate with a clinical information system for the execution of guidelines encoded in an enhanced format of GLIF2, the second version of GLIF⁷. During the years, the limitations of GLIF2 have been overcome and new requirements for guideline modeling have been included, resulting in the third version of GLIF, the GLIF3⁸. In this paper, we present our approach to implement the GLIF3 execution engine (GLEE) that tries to balance the requirement of sharability, flexibility and maintainability in guideline implementation.

SYSTEM ARCHITECTURE

We presume that GLEE will be integrated with a clinical information system as a middleware in the overall clinical information system architecture. GLEE provides standard interfaces to the EMR at the back-end and the specific clinical applications at the front-end. The communications between GLEE and the EMR at the back-end will enable access of various resources in the local environment that are required by a guideline system, such as patient data and clinical events. The communications between GLEE and the specific clinical applications at the front-end, for example, a physician order entry system, will enable smoothly integration of decision support within a clinician’s workflow⁹. The overall system architecture in terms of the relationship among GLEE, the local EMR, and the clinical applications is shown in Figure 1.

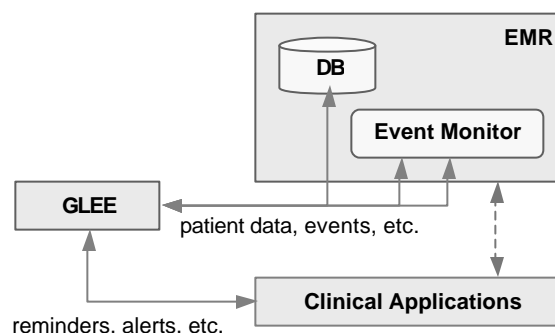


Figure 1. The relationship among GLEE, EMR and clinical applications

Within GLEE, we take a conceptually three-layer approach to clearly separate the functions provided by a guideline representation model (in our case, the GLIF3 model), the execution engine, and the hosting clinical information system. This introduces another interface between the guideline representation model and other components in the GLEE system. As part of the guideline representation model, this interface defines specific functions that need to be implemented by an execution engine.

As a guideline can be applied for multiple patients and a patient can be eligible for multiple guidelines, we decide to build GLEE as a client server system. This approach can facilitate the integration of GLEE with the local clinical information system and the communication between GLEE and the environment.

In our system architecture, guidelines encoded in GLIF3 format are stored in a guideline repository, which can be retrieved to the GLEE server. Currently, these guidelines are encoded using the Protégé-2000 knowledge acquisition tool¹⁰ and are exported as RDF¹¹ files. Execution records of a guideline applied to a specific patient, which we called traces, are also stored at the server side. We will discuss the tracing system in the next section.

Although we believe that GLEE's user interface should be integrated with a clinical application, we provide a standalone user interface at the client side to facilitate system development. This standalone user interface can also be used for education purpose or as a testing tool for guideline encoding. The internal structure of GLEE is shown in Figure 2.

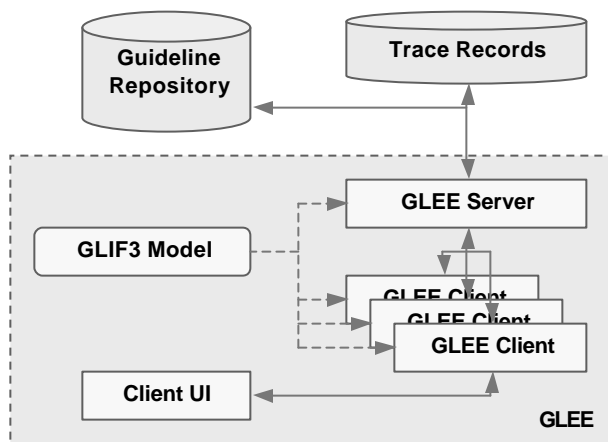


Figure 2. The internal structure of the GLEE

TASK SCHEDULING & TRACING SYSTEM

In GLIF3 model, guidelines are encoded as steps such as the decision step, the action step or the

patient state step that correspond to generic clinical tasks. Some type of step such as the branch step and the synchronization step are used for task scheduling. Steps can be chained together to form a flowchart-like algorithm that defines the temporal order of these steps during execution. Guidelines can be nested in GLIF3 to provide views with different granularities.

Traditional approach to schedule guideline execution is to mechanically find the executable tasks defined by an encoded guideline in the context of a specific patient. The whole process is completely controlled by the execution engine. A major drawback of this approach is that an encoded guideline could not address every possible clinical scenario, leading to the discrepancy between what the guideline system suggests and what a patient should really be. Several models^{8,12,13}, including GLIF3, have addressed this issue by providing representation primitive such as patient scenario or patient state that can be used to record a patient's clinical status in a specific context of guideline. These patient scenarios or patient states can then be used as entrances or exits of a guideline when applied to a patient. Although this solution at the representation level can provide some flexibility in execution, it still depends to a large extent on guideline encoders' enumeration of all possible entrance/exit points in a guideline.

Providing extra level of execution flexibility is one of the major considerations when we develop the GLEE system. In addition to the flexibility provided by GLIF3 at the representation level through the patient state step, we decide that GLEE should provide more flexibility at the execution level, with the user of GLEE as the final decision maker in task scheduling. In other words, at any time during the execution of a guideline, a user can follow the task schedule suggested by the system, or she can start or stop the execution of a specific step by her own judgment.

To distinguish the scheduled steps suggested by the system from those actually being executed by users' decisions, we use four states to represent the status of a guideline step during execution. These four states in execution include *prepared*, which means a step is suggested as executable by the execution system, *started*, which means a step is actually started by the user, *stopped*, which means a step has been intentionally stopped by the user before it is started or completes its execution, and *finished*, which means a step has normally finished its execution.

Typically, the GLEE scheduler suggests executable steps and put them into prepared states. This suggestion is based on the execution schedule encoded within a guideline or, in the case of a new encounter, based on the trace that records the

previous encounter with the guideline by current patient. A user can either confirm GLEE's suggestion on the execution schedule, or she can decide to override it by stopping a prepared step and starting another step she thinks to be appropriate. A user can also stop a started step to avoid unnecessary waiting for completion of the execution that is no longer relevant. If there is no manual interference from a user, GLEE will decide when a started step will finish its execution. Usually, this will trigger the execution of other steps that will be put by the GLEE scheduler into prepared states. The execution states and transitions in the GLEE system are shown in Figure 3. A screenshot of the algorithms, the guideline steps and their execution states from the standalone user interface is shown in Figure 4.

To keep the trace of guideline execution applied to a specific patient is an important feature of GLEE. This trace can be used as a hint for task scheduling at different encounters of guideline execution, or it can be used as a complete record for quality assurance to

inspect whether the cares provided to a patient are in compliance with guideline recommendations. Although some part of the trace can typically be found in a patient's medical record, many other parts may not have their correspondents. For this reason, we decide to keep an independent trace record system that can be used by the GLEE server, as shown in Figure 2. Currently, these trace records are implemented as XML files stored at the server side.

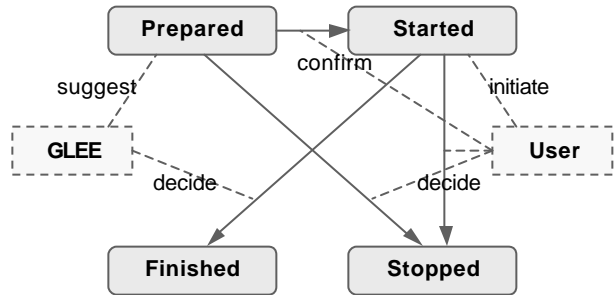


Figure 3. The execution states and transitions

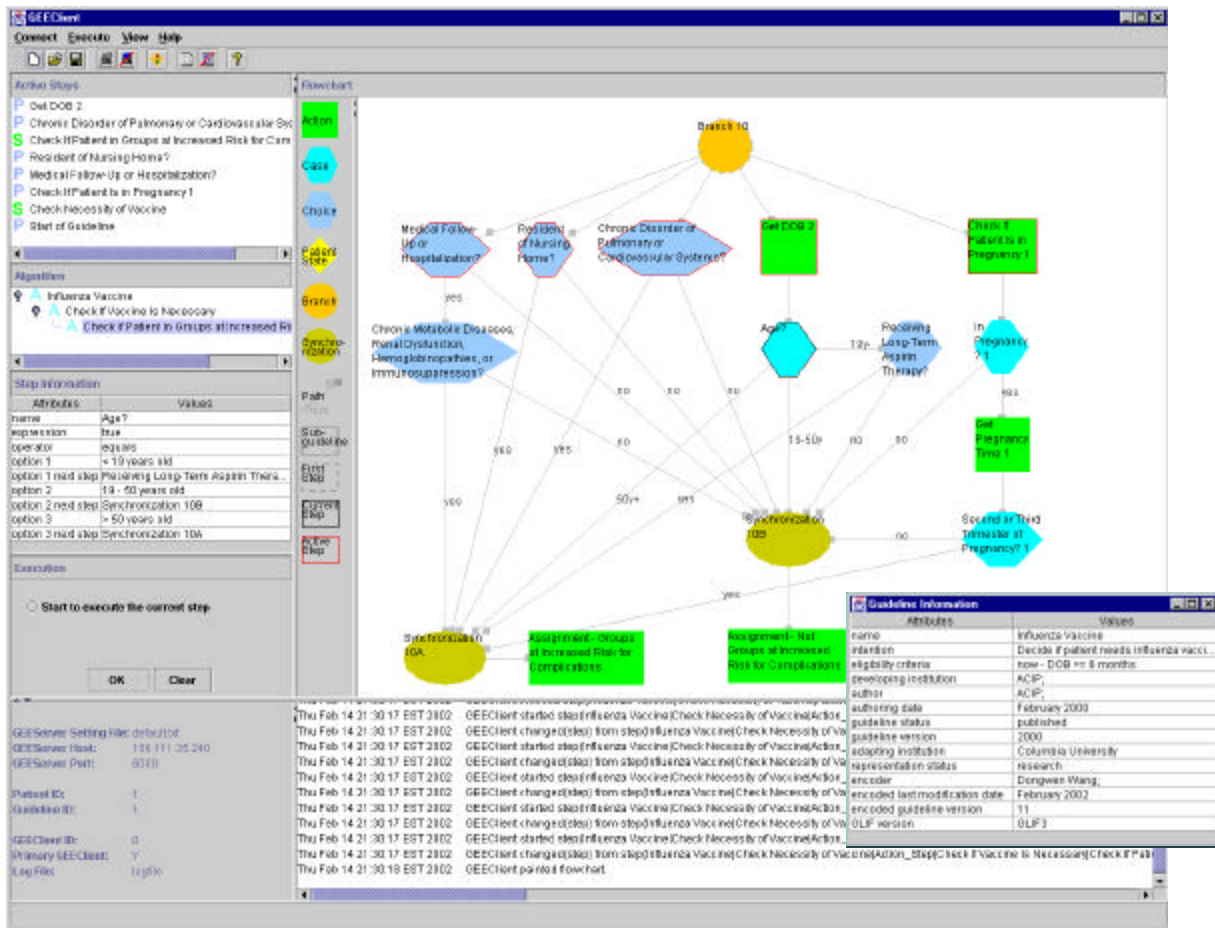


Figure 4. A screenshot of the algorithm, the guideline steps and their execution date states

EVENT MODEL

Event-driven execution model is an intrinsic part of clinical decision support systems¹⁴. GLIF3 supports this model by the definition of triggering events in a specific guideline step. As we take GLEE as a middleware that should be integrated with a local clinical information system, the event monitor, which is inherently embedded within a local system, will sit outside of GLEE. For this reason, we assume that there is an event monitor within the local clinical information system and the GLEE server will be able to use the service provided by it.

We define two types of events in the GLEE system. The clinical events are those events with clinical significance, for example, newly arrived lab test results or physician orders. Monitoring of these events depends on the implementation of the local clinical event monitor, with which we provide standard interface for communications. The system events are those internal events within the GLEE system that are used for execution purpose, for example, completion of a step preceding a synchronization step, which is used to decide whether the continuation criteria can be satisfied in the GLIF3 model. Monitoring of these events is an internal task and does not need the outside clinical event monitor.

When a step with a triggering event is started, GLEE will register that event, along with the current patient and guideline, in an internal event registration record. If the registered event is a clinical event, the GLEE server will send a message to the clinical event monitor in the local system for event registration. After that, the step will wait for the happening of the registered event to trigger its execution. If a clinical event happens, the clinical event monitor in the local system will send a message to the GLEE server to notify an event triggering. The GLEE server will then search the internal event registration record and find the corresponding patient, guideline, and guideline step that are waiting for the event. Consequently, that specific step will be triggered and start its execution. At the same time, the internal event registration record is adjusted to reflex the triggering of the registered event.

EXPRESSION LANGUAGE & PATIENT DATA

Expression language is an important component of a guideline representation model. In GLIF3, expression language is used to encode decision criteria and patient state.

GLIF3 can support different expression languages. In current implementation of GLEE, we use the

Guideline Expression Language (GEL) that is based on the logical expression of the Arden Syntax⁵ to encode decision criteria. Detailed discussion of GEL can be found elsewhere¹⁶.

Access of patient data is a critical task in guideline execution. For guidelines to be shared across different institutions, a standard data encoding system and a generic patient data model are two prerequisites. In recent years, various controlled medical terminologies have been developed as standards for data encoding. Unfortunately, there is no current consensus in the research community on a common patient data model. Thus, we do not assume any standard patient data model in our current implementation of GLEE. Instead, patient data access is through an interface to the local patient database, with the terminology concept and the data model class as the parameters that should be mapped to the local system.

INTERFACES TO HOSTING SYSTEMS

As discussed in the system architecture section, we provide several interfaces between GLEE and the local clinical information system. These interfaces are used as standard templates of communication for various purposes. Patient data access and event registration/triggering are two examples of these communications. We also provide a message function that can be used to send to the hosting system a generic message, the semantics of which can be defined locally. At the front-end, the interface for the communication between GLEE and the specific clinical applications can be used to provide services, such as reminders or alerts, to the user of a clinical decision support system. In general, these interfaces provide the flexibility to integrate GLEE to a hosting clinical information system.

DISCUSSIONS

Usually, the decision support provided by a guideline system is presented to the clinicians through alerts or reminders. However, it is the clinicians that will finally decide whether they will follow what the system suggests. The flexibility at the execution level provided by the GLEE system exactly addressed this mode of use in practice. This approach has overcome the limitation of previous approaches that depends on guideline encoders' enumeration of a patient's all possible clinical state in the context of guideline application.

Integration of decision support within clinicians' workflow is a critical factor for its success⁹. Thus, we

believe a standard interface between a guideline execution system and specific clinical applications is important for guideline implementation in clinical settings. The interface to the clinical applications provided by the GLEE system will hopefully facilitate this integration and promote the use of guidelines in clinical practice. Nevertheless, we do not exclude the possibility that a local system can provide its own methods of communication and presentation of alerts and reminders. We thus provide the messaging function at the back-end that can be used in this case as an alternative to facilitate the communication and integration. Finally, the interface between the guideline representation model and the other parts of the execution engine will facilitate the maintenance of the execution system itself, such as the evolution of the representation model.

We envision that the potential use of the GLEE system as a tool includes, a) in clinical decision support systems to provide assistance of decision-making in practice, b) for quality assurance to examine the compliances with care standards, c) in guideline development to investigate whether the encoded guideline faithfully reflexes a guideline encoder's intention, and d) in medical education to provide an imitate environment for guideline application. However, we understand that wide acceptance of guideline systems used in clinical practice depends also on many other factors, such as development of a generic patient data model and in-depth understanding of local adaptation of guidelines, which are not addressed in our current work.

Acknowledgement

This work is supported by grant LM06594 to the InterMed Collaboratory from the National Library of Medicine, the Department of the Army, and the Agency for Healthcare Research and Quality. We gratefully acknowledge Mor Peleg, Samson Tu, Aziz Boxwala, Omolola Ogunyemi, Qing Zeng, Robert Greenes, Vimla Patel, and other people in the InterMed Collaboratory for their contributions in the design of the GLEE system, the development of the GLIF3 model, and implementation of the GEL language.

References

1. Institute of Medicine. Guidelines for clinical practice: from development to use. Washington, DC: National Academy Press; 1992.
2. McDonald CJ, Overhage JM. Guidelines you can follow and trust: an ideal and an example. *JAMA*. 1994;271:872-3.
3. Johnston ME, Langton KB, Haynes RB, Mathieu A. Effects of computer-based clinical decision support systems on clinician performance and patient outcome. *Ann Intern Med*. 1994;120:135-42.
4. Tu SW, Musen MA. Representation formalisms and computational methods for modeling guideline-based patient care. *First European Workshop on Computer-Based Support for Clinical Guidelines and Protocols*. 2000;:125-42.
5. Wang D, Peleg M, Tu SW, Shortliffe EH, Greenes RA. Representation of clinical practice guidelines for computer-based implementations. *Medinfo*. 2001;10:285-9.
6. Ohno-Machado L, Gennari JH, Murphy S, Jain NL, Tu SW, Oliver DE, et al. The GuideLine Interchange Format: a model for representing guidelines. *J Am Med Inform Assoc*. 1998;5(4):357-72.
7. Boxwala AA, Greenes RA, Deibel SR. Architecture for a multipurpose guideline execution engine. *Proc AMIA Symp*. 1999;:701-5.
8. Peleg M, Boxwala AA, Ogunyemi O, Zeng Q, Tu S, Lacson R, et al. GLIF3: The evolution of a guideline representation format. *Proc AMIA Symp*. 2000;:645-9.
9. Tierney WM, Overhage JM, McDonald CJ. Computerizing guidelines: factors for success. *Proc AMIA Annu Fall Symp*. 1996;:459-62.
10. Noy NF, Fergerson RW, Musen MA. The knowledge model of Protégé-2000: Combining interoperability and flexibility. *The 2nd International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000)*. 2000.
11. Resource Description Framework (RDF). Available at: <http://www.w3.org/RDF/>. Accessed February 9, 2002.
12. Tu SW, Musen MA. A flexible approach to guideline modeling. *Proc AMIA Symp*. 1999;:420-4.
13. Johnson PD, Tu S, Booth N, Sugden B, Purves IN. Using scenarios in chronic disease management guidelines for primary care. *Proc AMIA Symp*. 2000;:389-93.
14. Hripcsak G, Clayton PD, Jenders RA, Cimino JJ, Johnson S. Design of a clinical event monitor. *Comput Biomed Res*. 1996;29(3):194-221.
15. Hripcsak G, Ludemann P, Pryor TA, Wigertz OB, Clayton PD. Rationale for the Arden Syntax. *Comput Biomed Res*. 1994;27(4):291-324.
16. Peleg M, Ogunyemi O, Tu S, Boxwala AA, Zeng Q, Greenes RA, et al. Using features of Arden Syntax with object-oriented medical data models for guideline modeling. *Proc AMIA Symp*. 2001;:523-7.