

Guideline Execution Engine (GEE) Specification

Philosophy of Design

We try to realize several levels of reuse.

Reuse of the GLIF3 Model

We provide an execution task interface for the GLIF3 model so that it is possible to realize different implementations of a Guideline Execution Engine (GEE).

For this purpose, we distinguish tasks required by a specific guideline model for proper execution from tasks required by a GEE for administrative purpose.

We provide a set of APIs (Guideline Model Interface) for the implementation of a GEE to support execution of guidelines encoded in GLIF3. This Guideline Model Interface can be considered as an attachment of the GLIF3 model to regulate how the elements in GLIF3 can be used. In other words, they define the semantics of GLIF3. In object-oriented jargon, these are the methods, comparing to current elements in the GLIF3 model, which are attributes.

Institutions can implement their own GLIF3 GEEs with appropriate interfaces to the Guideline Model Interface. We provide one implementation of GLIF3 GEE based on the Guideline Model Interface.

Reuse of our GEE

Our implementation of the GEE is a specific implementation based on the Guideline Model Interface attached with the GLIF3 model. We try to realize reuse of this specific GEE in different institutions.

For this purpose, we distinguish local system dependent components from other components that can be shared by different local systems.

We provide a set of APIs (Vocabulary and RIM, GEE Server EMR Interface, and Guideline Repository Interface at the server side, and Guideline Model Interface at the client side) that connect the local system dependent components to the other core components of the GEE. Institutions will implement their own local system dependent components of the GEE according to the APIs provided.

System Architecture of GEE

Assumptions in the design:

- Guidelines have been already loaded from the global guideline server to a local institution, adapted and stored in a local guideline repository.
- Patient data, actions, events, etc. can be mapped to a local EMR through the vocabulary and RIM used in the guideline model.

Design of our GEE takes the client-server system architecture. The system consists of one GEEServer and multiple GEEClients.

The GEEServer handles all “real” execution tasks as well as administrative tasks of the whole GEE. The GEEServer package consists of the local system independent components of the GEE. The GEEServerLocalSystem package consists of the local system dependent components of the GEE.

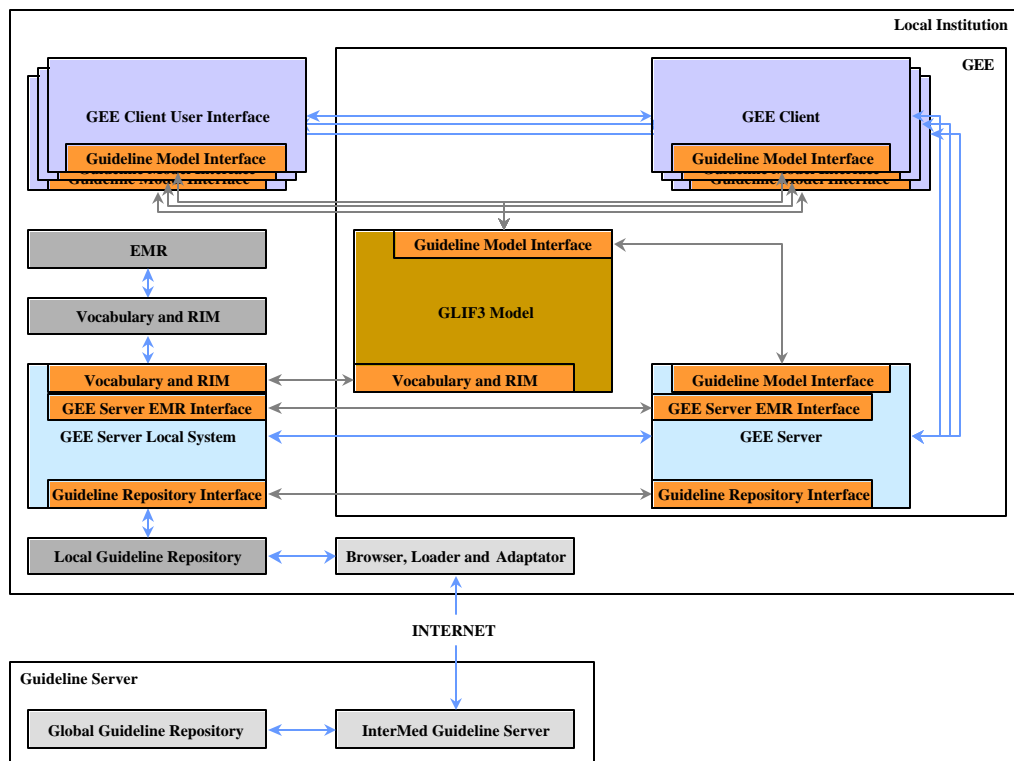
The GEEClients only provide user interfaces and their own administrative tasks. The GEEClient package is independent of the local system. The GEEClientUserInterface package depends on the local system. This part provides integration of the front end to a hosting system, such as an order entry system. Within each

GEEClient, a user can execute one guideline applied to a specific patient. Executions of multiple guidelines applied to multiple patients are realized through multiple launching of the GEEClient.

We provide a standalone client side user interface in the GEEClientUserInterface package.

Communication between the GEEServer and a GEEClient is two-way.

- A user's interaction with a GEEClient will be sent to the GEEServer. After appropriate processing, the result will be sent back to the GEEClient as feedback to the user. These communications are initiated by the GEEClients.
- When registered events and exceptions are triggered, the GEEServer can initiate communications with the GEEClients so that the triggered events and exceptions will be notified to the GEEClients.



Use Cases

Use cases are shown in the following diagram.

Actors related to the GEE Server include the GEE administrator, the guideline repository and the EMR.

- **GEE Server Initialization**
In this scenario, the GEE administrator starts the GEEServer. The GEEServer then loads guidelines from the guideline repository, parses them, builds the internal data structures, maps patient data, actions and events to local EMR, and starts the server daemon.
- **GEE Server Processing**
In this scenario, the GEEServer handles communication from GEEClients and the EMR (for actions, events and exceptions), processes them appropriately and provides feedbacks.
- **Event Triggering**

In this scenario, the GEEserver already registered events (initiated by clients) that need to be monitored. When these events happen, the EMR communicates with the GEEserver, appropriate processing are performed and results are feeded back to the users that are using the related guidelines for specific patients.

- GEE Server Termination

In this scenario, all GEEclients are already terminated. The GEE administrator can stop the server daemon and terminate the GEEserver.

Actors related to the GEEclients are the users.

- GL Execution Initialization

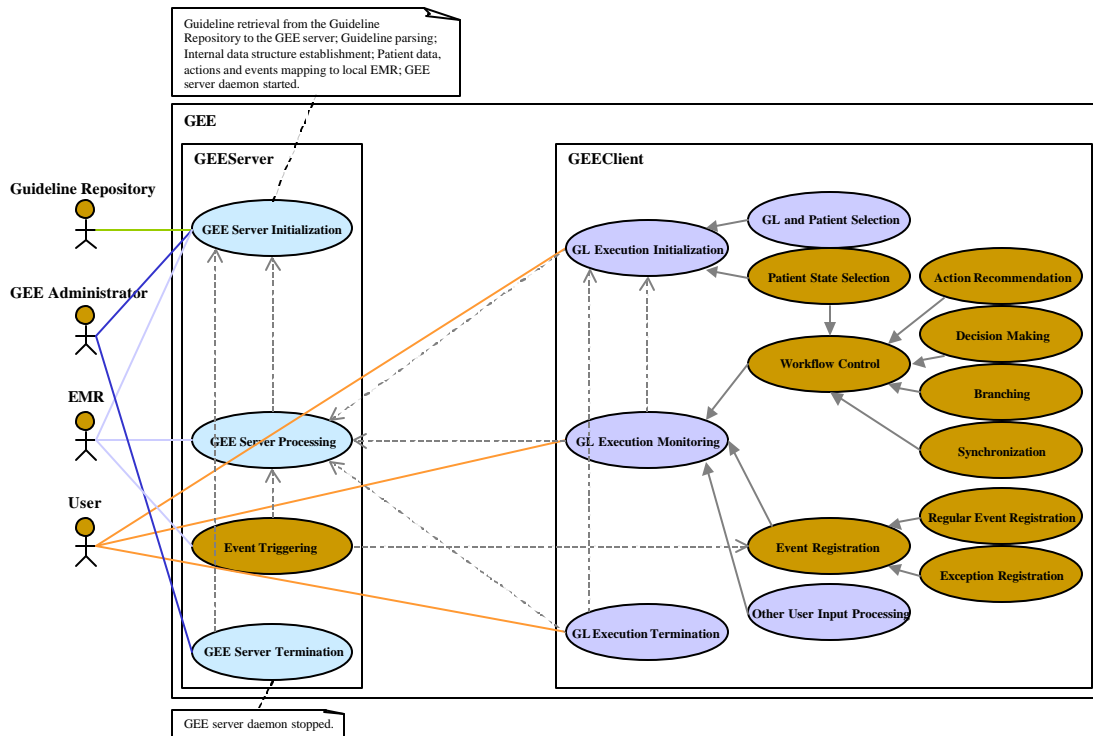
In this scenario, the user starts the GEEclient. The GEEclient then communicate with the GEEserver so that a GEEclient instance will be created at the server side. The user then can select a guideline to be executed and a patient to be applied. In case this is a returning session, the patient state applied is also decided.

- GL Execution Monitoring

Interactions between the user and the GEEclient include 1) workflow control, which further consists of patient state selection, action recommendation, decision making, branching and synchronization management, 2) event registration, which include regular event registration and exception registration, and 3) other user input processing, for example, manual confirmation of action or decision making.

- GL Execution Termination

The user completes the execution of a guideline or a session of it applied to a specific patient. The server side will destroy the GEEclient instance.

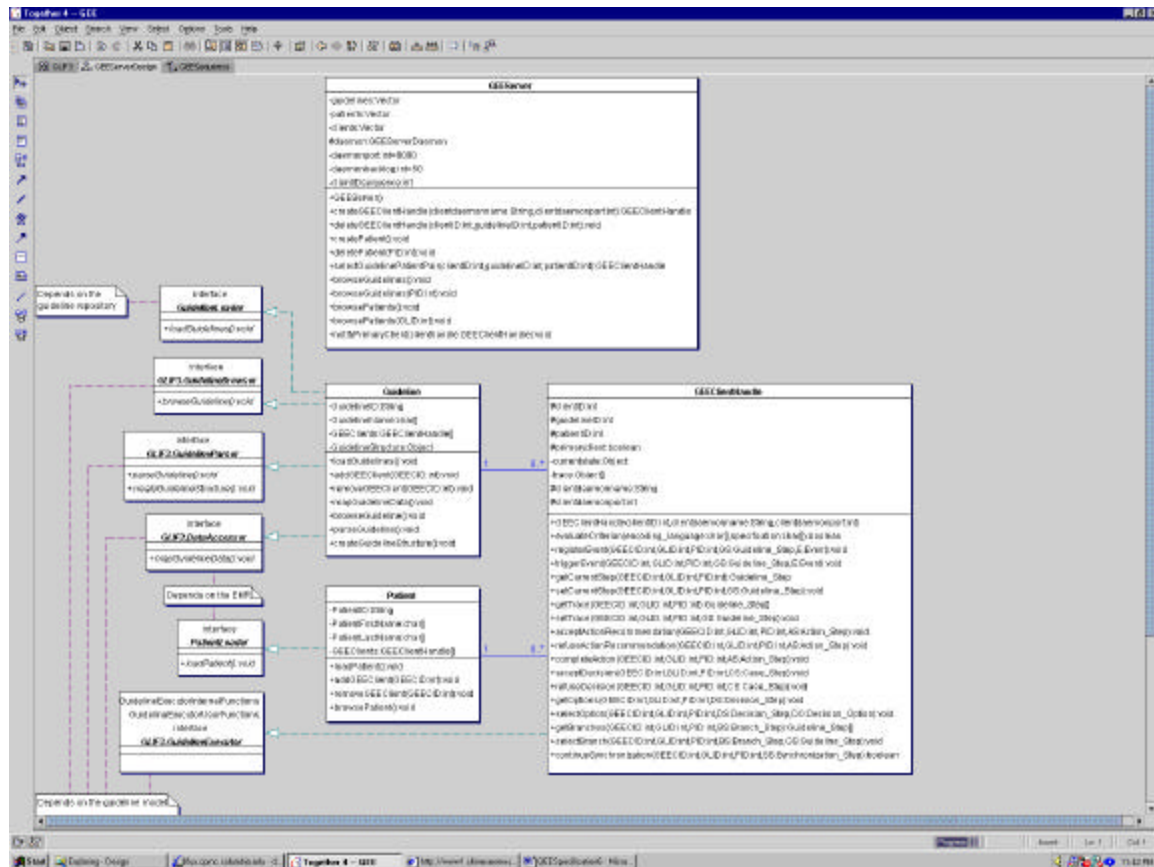


There are five packages in the system, GEEServer, GEEClient, GLIF3, GEEServerLocalSystem, and GEEClientUserInterface. All other four packages will import GLIF3.

- GEEServer

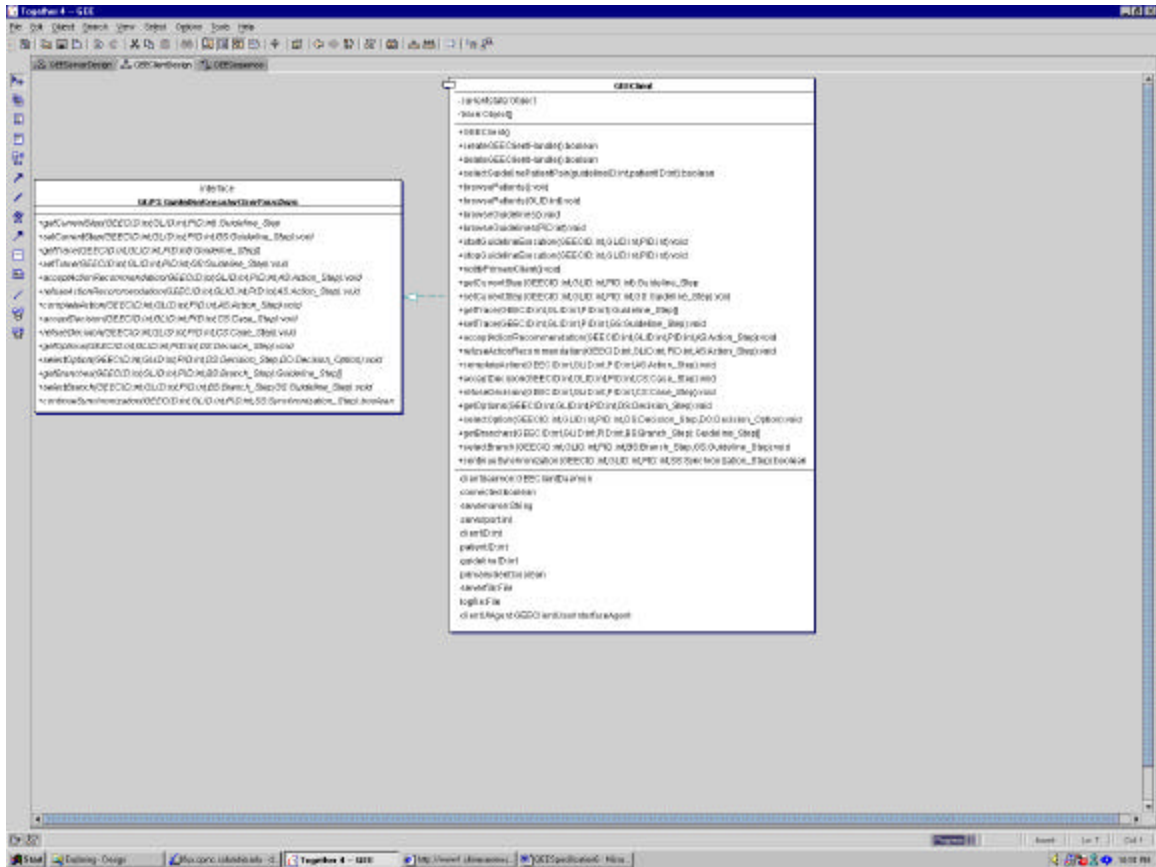
Major classes in the GEESServer package include GEESServer, GEEClientHandle, Patient and Guideline. Each GEEClientHandle instance corresponds to a specific guideline applied to a specific patient (which is a GEEClient instance in the GEEClient package at the client side).

Guideline model specific methods are wrapped in the GLIF3.GuidelineExecutor interface, which is implemented by the GEEClientHandle. Other interfaces whose implementations depend on the guideline model include GuidelineParser, GuidelineBrowser and DataAccessor. DataAccessor also depends on the EMR. Implementation of PatientLoader interface depends on the EMR. Implementation of GuidelineLoader interface depends on the guideline repository.



- GEEClient

Major class in the GEEClient package is the GEEClient. It needs to implement GLIF3.GuidelineExecutorUserFunction. The implementation by GEEClient focuses on the communication with GEEServer.

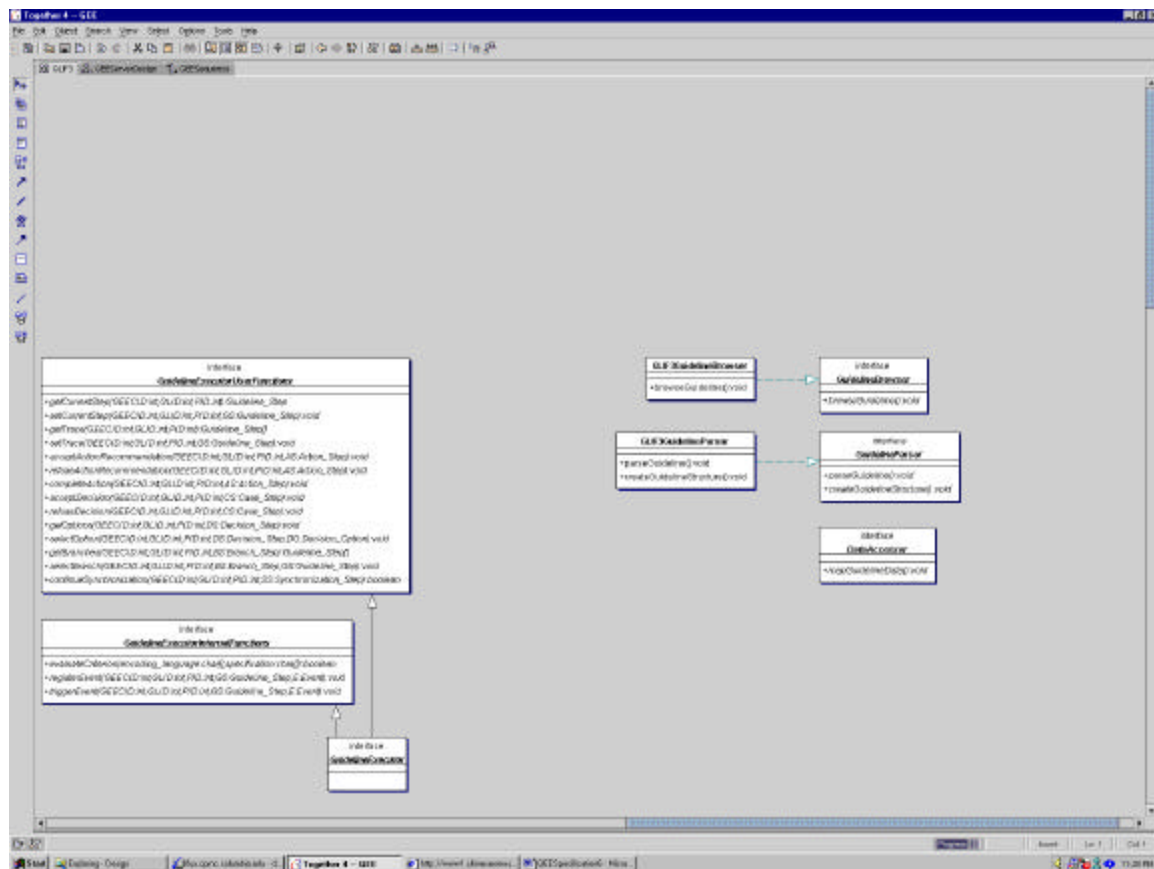


- GLIF3

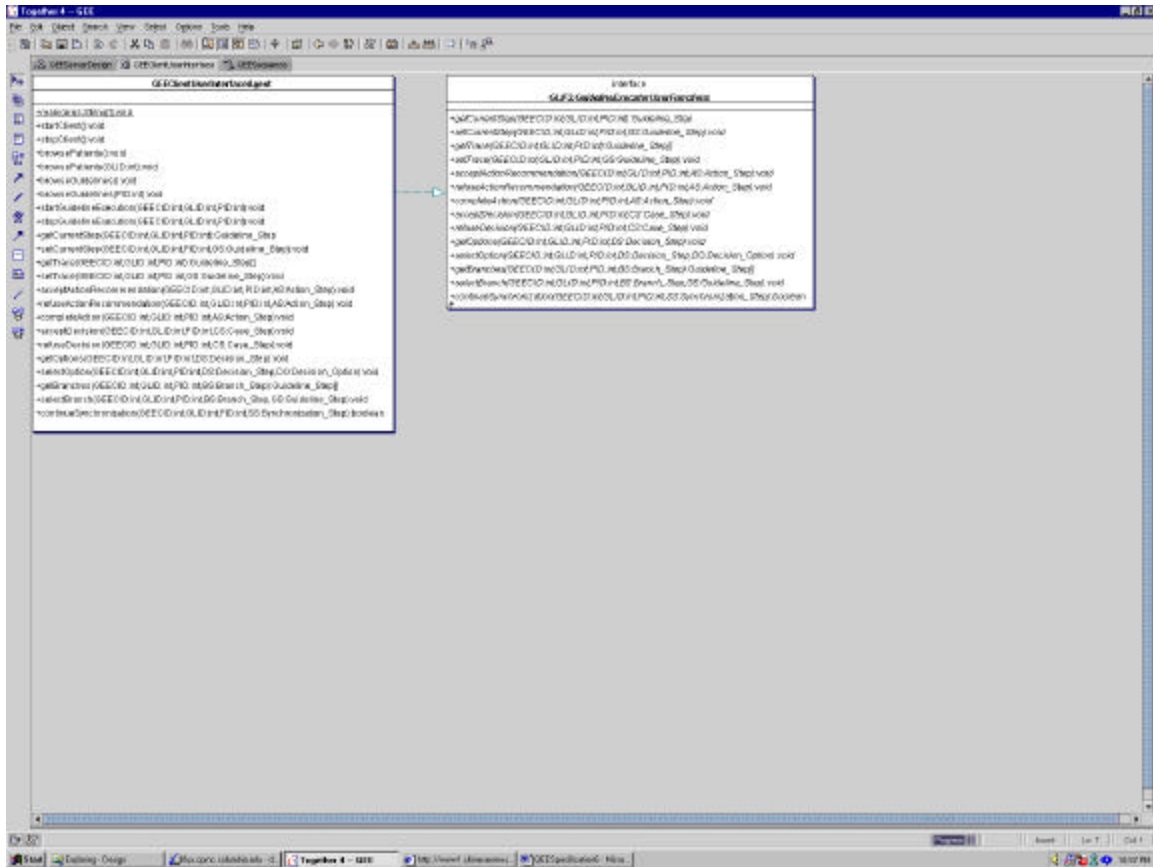
There are three major interfaces in the GLIF3 package, `GuidelineExecutorUserFunction`, `GuidelineExecutorInternalFunction` and `GuidelineExecutor`. The latest one extends the previous two. `GuidelineExecutorUserFunction` specifies the functions users directly used when execute GLIF3 guidelines, while `GuidelineExecutorInternalFunction` specifies the functions that need to be implemented by the GEE to support the user functions.

Other three interfaces, `GuidelineBrowser`, `GuidelineParser` and `DataAccessor` specify methods that will be used by `GEEServer`. `GLIF3GuidelineBrowser` and `GLIF3GuidelineParser` are two implementations of the first two interfaces specific to GLIF3.

The whole GLIF3 model elements should also be presented as classes in the GLIF3 package. The methods in the above interfaces will specify which classes and attributes will be used in the implementation. This defines the semantic of GLIF3. This work is in progress.



- Major class in the GEEClientUserInterface package is the GEEClientUserInterfaceAgent. It also needs to implement GLIF3.GuidelineExecutorUserFunction. The implementation by GEEClientUserInterfaceAgent focuses on the presentation at the client user interface.



- GEEServerLocalSystem

Major classes in the GEEServerLocalSystem package include MyGuidelineLoader, MyPatientLoader and MyDataAccessor, which separately implement GuidelineLoader, PatientLoader and DataAccessor interface. MyGuidelineLoader depends on the guideline repository. MyPatientLoader depends on the EMR. MyDataAccessor depends on both the EMR and the guideline model.

