Changes to GLIF Specification
September 9, 2002

1) Aziz: We will add triggering_event to patient_state_step
The specification will be revised to say that two triggers in the same step cannot have the same priority.

Mor: Each Trigerring_Event is associated with an earliest and latest start time. This means that once an event happens, you trigger an associated action/decision step after the earliest time constraint and before the latest time constraint. When more than one triggering event that is associated with one action/decision step fires at a given time, and they have different earliest and latest start time constraints, then the action/decision will occur at different times. Only one triggering event is chosen, based on the priority attribute.

Dongwen: Say, there are two events. Event A has a lower priority, but earlier earliest_start_time and latest_start_time. Event B has a higher priority, but later earliest_start_time and latest_start_time. Suppose even the latest_start_time of event A is earlier than the earliest_start_time of event B. Does this mean if both A and B have been triggered, we just ignore A and only consider B? It seems to me in this case we still need to take event A into account as its time constraint is not overlapped with that of event B.

Mor: The question is: do you want to invoke a step twice if two triggering events happen at the same time. I think that you would only want to trigger the step once. The question is, do you want to trigger the step as soon as possible, or according to the event with the highest priority? I don't know. We need to come up with a real-life example to answer this.

Do we really want to get into events and exceptions? When I first introduced these concepts there was very little interest in it. As a result, it was not thoroughly validated, and we decided to put it in the appendix of the Tech Spec.

Dongwen: If the triggering of events are associated with temporal constraints, they will become very complex. Also I believe it depends to a large extent on whether the local clinical information system supports this type of "real-time" execution. I agree it may be more appropriate to put them in the appendix.

2) Aziz: We will merge choice and case step classes into one class: Decision_Step. Decision_Step will no longer be abstract. Decision_Step will be structured similar to the current Choice_Step. We can use strict rule in to encode the decision criterion of the case step. A Boolean attribute called automatic will distinguish between manual and automatic decisions. We will add an attribute called default_option_when_automatic that points to the default option when the step is in automatic mode.

The distinction between a user-choice and automatic-selection is made during local implementation. By making the distinction at guideline-design-time, we will force local implementers to remodel the guideline by deleting one step and replacing it with another. The simpler thing for them to do would be to check or uncheck the box, which says the decision is automatic or not. PROforma follows a similar paradigm.

3) Dongwen: How do we aggregate the rule ins and outs?

Mor: How about a very simple aggregation: if strict rule out applies: this option is ruled out….

Dongwen: There may exist multiple strict rule out criteria. Any one applies will rule out the option?

Mor (continued): …give one point for every rule-in that applies. Deduct one point for every rule out. Give 5 points for every strict rule in. It is very arbitrary, but can demonstrate an aggregation function.

Dongwen: If strict rule in is still counted by points, I think we may also count the strict rule out by points to make the system looks consistent.

What about the overall aggregation function? The total points >= 0 will be rule in, and < 0 will be rule out?

Another thought on this is that this type of ranking and aggregation function may be more appropriate when it is taken and used in a local institution. But it is hard to share as acceptance of this function by different institutions may be hard.

4) Aziz: The message_action class will be removed from the action_specification hierarchy. This seems like a level C kind of thing. The Medically_oriented_action will be used to specify text recommendations, instead of the message_action class

5) Aziz: Get_Data_Action needs to be modified for use with Gello. We will drop the attributes used for pulling out specific primitive data items. In order to continue supporting gel, we can create a gel-compatible subclass called Get_Data_Action_Gel.