Clinical Guidelines Adaptation: Managing Authoring and Versioning Issues

Paolo Terenziani[1], Stefania Montani[1], Alessio Bottrighi[1], Gianpaolo Molino[2], Mauro Torchio[2]
[1] DI, Univ. Piemonte Orientale, Via Bellini 25/G, Alessandria, Italy
{terenz, stefania, alessio}@mfn.unipmn.it
[2] Lab. Informatica Clinica, Az. Ospedaliera S. G. Battista, C.so Bramante 88, Torino, Italy
{mtorchio, gmolino}@molinette.piemonte.it

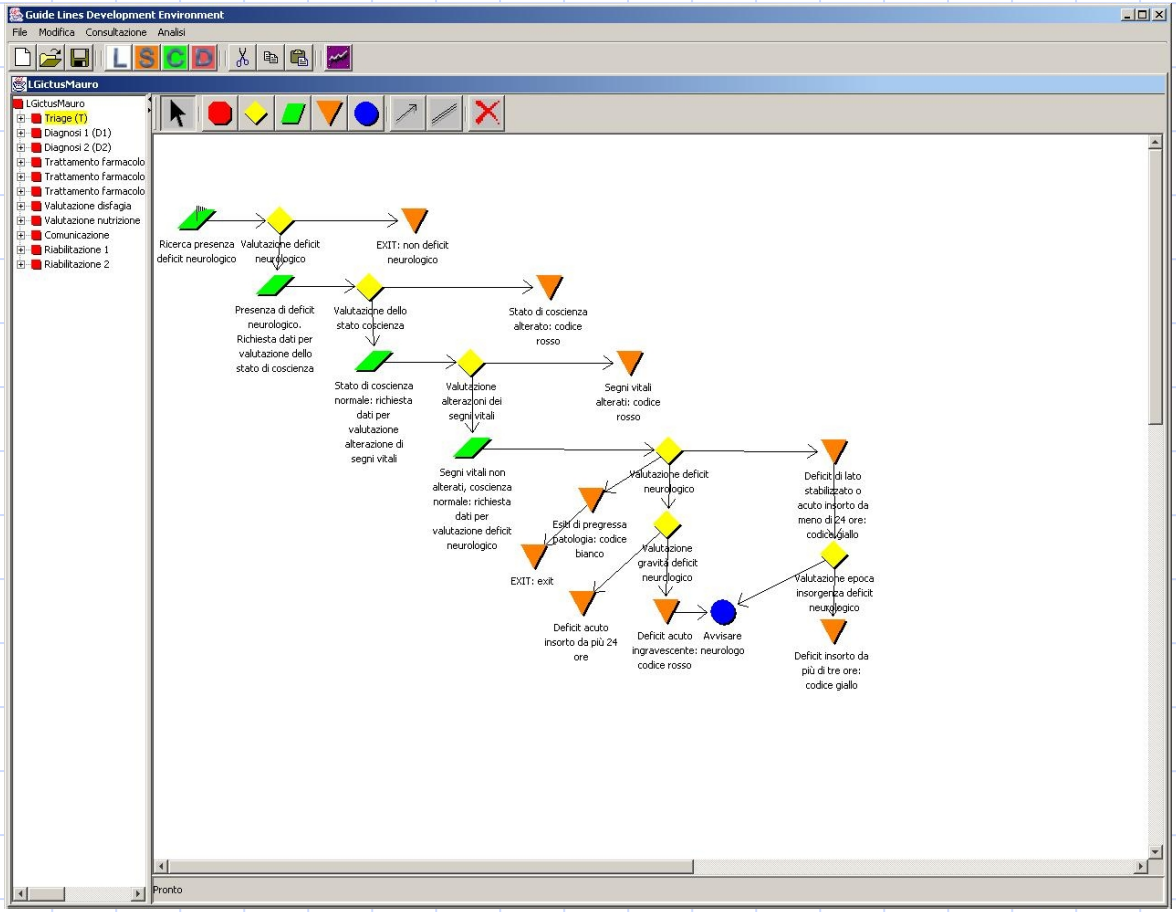
GLARE (GuideLine Acquisition Representation and Execution)

- Joint project:
•Dept. Comp. Sci., Univ. Alessandria (It): P. Terenziani, S.Montani, A.Bottrighi
•Dept. Comp. Sci., Univ. Torino (It): L.Anselma,G.Correndo
•Az. Osp. S. Giovanni Battista, Torino (It): G.Molino, M.Torchio

-Domain independent (e.g., bladder cancer, reflux esophagitis, heart failure)

-User-friendly (limited number of primitives)

GLARE (Guideline Representation)



GLARE (main features)

Acquisition Tool

-Facilities for syntactic & semantic consistency checking

-Temporal constraints

Execution Tool

-Facilities for decision making (hypothetical reasoning facility)

-Temporal constraints

Interaction with HIS

- during acquisition
- during execution

Testing: bladder cancer, reflux esophagitis, heart failure, ischemic stroke

Guideline Adaptation

Adaptation needed for dissemination and use

Local resources adaptation [Medinfo 04]

Local software environment adaptation [Medinfo 04, CGP 04]

Update adaptation

"Cultural" adaptation

Update Adaptation

Two orthogonal issues:

(1) History of the different **versions** (e.g. to justify physicians' past decisions)

(2) Management of authors (users vs. supervisors)
(to model update proposal vs update validation)

Guideline Adaptation

(1) management of authors (users vs supervisors);

(2) management of the status of any piece of knowledge in the GL (proposed vs. accepted knowledge)

(3) management of the history of knowledge (considering proposal vs acceptance times);

(4) facilities for selecting the parts of a guideline to be changed/updated;

(5) facilities for modifying (part of) a guideline

(6) facilities for formulating queries and for visualising the updates.

Three-layered approach

(1) A data model layer, defining the data model and providing the basic operations;[AIME'05]

(2) query language layer, supporting an SQL-like high-level manipulation and query language, based on layer (1);

(3) an interface layer, based on the previous ones, that provides users with high-level functionalities, accessed through a user-friendly graphical interface

Data model: an example

A {TP: **<t_0,S1>** # TA: **<t_0,S1>**}

P=v {TP: **<t_0,S1>** # TA : **<t_0,S1>**}

A1 {TP: **<t_0,S1>** TA: **<t_0,S1>**}

P1=v1 {TP: **<t_0,S1>** # TA: **<t_0,S1>**}

A2 {TP: **<t_0,S1>**,<t_3,U2> # TA: **<t_0,S1>**}

P2=v2 {TP: **<t_0,S1>** # TA: **<t_0,S1>**}

(0) Initial situation at the time t_0 of acquisition (**bold**). Action A, which has a property P with value v, and is composed by A1 (with property P1 with value v1) and A2 (with property P2 having value v2).

Data model: an example

A {TP: <t_0,S1>,**<t_1,U1>** # TA: <t_0,S1>}

P=v {TP: <t_0,S1> # TA : <t_0,S1>}

A1 {TP: <t_0,S1>,**<t_1,U1>** # TA: <t_0,S1>}

P1=v1 {TP: <t_0,S1>,**<t_1,U1>** # TA: <t_0,S1>}

P1=v1' {TP: <t_1,U1> # TA :}

A2 {TP: <t_0,S1> # TA: <t_0,S1>}

P2=v2 {TP: <t_0,S1> # TA: <t_0,S1>}

(1) At time t_1, U1 proposes to change the value of property P1 to v1'.

Data model: an example

A {TP: $<t_0,S1>$,$<t_1,U1>$ # TA: $<t_0,S1>$,**$<t_2,S1>$**}

P=v {TP: $<t_0,S1>$ # TA : $<t_0,S1>$}

A1 {TP: $<t_0,S1>$,$<t_1,U1>$ # TA: $<t_0,S1>$,**$<t_2,S1>$**}

P1=v1 {TP: $<t_0,S1>$,$<t_1,U1>$ # TA: $<t_0,S1>$}

P1=v1' {TP: $<t_1,U1>$; # TA **:** **$< t_2,S1>$**}

A2 {TP: $<t_0,S1>$ # TA: $<t_0,S1>$}

P2=v2 {TP: $<t_0,S1>$ # TA: $<t_0,S1>$}

(2) At time t_2, the supervisor S1 accepts the above update.

Data model: an example

A {TP: $<t_0,S1>$,$<t_1,U1>$,**$<t_3,U2>$** # TA: $<t_0,S1>$,$<t_2,S1>$}

P=v {TP: $<t_0,S1>$ # TA : $<t_0,S1>$}

A1 {TP: $<t_0,S1>$,$<t_1,U1>$ # TA: $<t_0,S1>$,$<t_2,S1>$}

P1=v1 {TP: $<t_0,S1>$,$<t_1,U1>$ # TA: $<t_0,S1>$}

P1=v1' {TP: $<t_1,U1>$; <TA : $< t_2,S1>$}

A2 {TP: $<t_0,S1>$,**$<t_3,U2>$** # TA: $<t_0,S1>$}

P2=v2 {TP: $<t_0,S1>$ # TA: $<t_0,S1>$}

P3=v3 {TP: **$<t_3,U2>$**#TA :}

(3) At time t_3, U2 proposes to insert a new property P3 to A2, with value v3.

Data model: an example

A {TP: $<t_0,S1>,<t_1,U1>,<t_3,U2>$ # TA: $<t_0,S1>,<t_2,S1>,$**$<t_4,S1>$**}

P=v {TP: $<t_0,S1>$ # TA : $<t_0,S1>$}

A1 {TP: $<t_0,S1>,<t_1,U1>$ # TA: $<t_0,S1>,<t_2,S1>$}

P1=v1 {TP: $<t_0,S1>,<t_1,U1>$ # TA: $<t_0,S1>$}

P1=v1' {TP: $<t_1,U1>$; <TA : $< t_2,S1>$}

A2 {TP: $<t_0,S1>,<t_3,U2>$ # TA: $<t_0,S1>,$**$<t_4,S1>$**}

P2=v2 {TP: $<t_0,S1>$ # TA: $<t_0,S1>$}

P3=v3 {TP: $<t_3,U2>$#TA : **$<t_4,S1>$**}

(4) Finally, the acceptance at time t_4 of the proposal in t_3 leads to the final situation.

Query language

Select {selection clause – default: empty}

From {guideline identifier}

Where {selection conditions – default: empty}

STATUS {proposed/accepted – default: accepted}

AS_BY {authors – default: empty }

AS_IN {point / period / alltime → default: point=NOW}

Query language: an example

Q1: Give me the current description of A2.

STATUS={accepted}; AUTHOR={any Supervisor}; TIME=NOW;

A2.P2=v2 A2.P3=v3

Q2: Give me the description of A2 at time t_0.

STATUS={accepted}; AUTHOR={any Supervisor}; TIME= t_0;

A2.P2=v2

Q3: Give me the update proposals regarding A made by U1 or U2 in the period $[t_0,t_2]$

STATUS={proposed}; AUTHOR={U1,U2}; TIME=$[t_0,t_2]$;

A1 part-of A; A1.P1=v1 ➔ A1.P1=v1'

Note. The updates to A1 are taken into account since A1 is part of A.

Q4: Give me the history of (acceptances regarding) A

STATUS={accepted}; AUTHOR={ any Supervisor}; TIME= "all-time"

At time t_0: A.P=v A1 part-of A; A1.P1=v1 A2 part-of A; A2.P2=v2

At time t_2: A1 part-of A; A1.P1=v1 ➔ A1.P1=v1'

At time t_4: A2 part-of A; added A2.P3=v3