

Analysis of the GLARE and GPROVE Approaches to Clinical Guidelines

Alessio Bottrighi¹, Federico Chesani², Paola Mello², Marco Montali²,
Stefania Montani¹, Sergio Storari³, and Paolo Terenziani¹

¹DI, Univ. Piemonte Orientale “A. Avogadro”, Alessandria, Italy

²DEIS, Univ. Bologna, Bologna, Italy

³ENDIF, Univ. Ferrara, Ferrara, Italy

{alessio, stefania, terenz}@mfu.unipmn.it,
{federico.chesani, marco.montali, paola.mello}@unibo.it,
sergio.storari@unife.it

Abstract. Clinical guidelines (GLs) play an important role in medical practice, and computerized support to GLs is now one of the most central areas of research in Artificial Intelligence in medicine. In recent years, many groups have developed different computer-assisted management systems of GL. Each approach has its own peculiarities and thus a comparison is necessary. Many possible aspects can be analyzed, but a first analysis has probably to consider the GL models, i.e. the representation formalisms provided. To this end, Peleg and al. [4] have analyzed and compared six different frameworks. In this paper, we analyse also GLARE and GPROVE on the basis of the same methodology. Moreover, we extend such analysis by considering the tools and the facilities that GLARE and GPROVE provide to support the use of GLs. The final goal of our analysis is to exploit the differences between these two systems and if they can be fruitfully integrated.

Keywords: clinical guideline, computer-assisted guideline manager, guideline model, decision support, verification.

1 Introduction

Clinical guidelines (GLs) represent the current understanding of the best clinical practice. In recent years the importance and the use of GL are increasing in order to improve the quality and to reduce the cost of health care. Many different systems and projects have therefore been developed in order to realise computer-assisted management of GL (see, e.g., the collections [1-3]), and computer-based GL management is now one of the most central areas of research in Artificial Intelligence (AI) in medicine and in medical decision making.

From the point of view of the GL model provided by such systems, i.e. of the GL representation language, a comparison among some existing systems is described in [4]. Such analysis concerns six approaches: Asbru [5], EON [6], GLIF [7], [8], GUIDE [9], PRODIGY [10], and PROforma [11].

Since 2003, new approaches have been developed (eg. GPROVE[12], HeCaSe [13], Helen[14], SpEM[15]) and some others (e.g. GLARE[16], GASTON [17], SAGE[18]) was not considered in [4]. In this work, we extend this comparison by analysing also GLARE [16] and GPROVE [12]. The methodology described in [4] concerns mostly a review of syntactic features. Since we consider important the analysis of the GL models, on the other hand we consider relevant that a computer-based GL system supports the user-physicians. Thus we address the tools and verification techniques provided by GLARE and GPROVE as well.

GLARE (Guideline Acquisition, Representation and Execution) [16] is a domain-independent prototypical system to acquire, represent and execute GL, which has been built starting from 1997 by University of Piemonte Orientale in cooperation with Azienda Ospedaliera San Giovanni Battista in Turin, one of the largest hospitals in Italy, and has been successfully tested in different domains, including bladder cancer, reflux esophagitis, and heart failure.

GPROVE (Guideline PRocess cOnformance VERification framework) [19] is a set of tools for the specification and run-time/a-posteriori compliance verification of the observed GL behaviour. It is composed by a graphical process definition language called GOSpeL (Guideline prOcess Specification Language), by an automatic mapping/translation module towards a formal language called *SCIFF* [20], and by an operational counterpart (a proof procedure) of the *SCIFF* formalism, that is used to verify the compliance of a given execution with respect to the defined GL process. GPROVE has been built by University of Bologna and University of Ferrara, and, in the SPRING PRRITT project sponsored by Emilia Romagna region, it has been successfully tested with the Cancer Screening Guideline adopted by the sanitary organization of the Emilia Romagna region.

Main goal of our work is to study the feasibility of integrating GLARE and GPROVE in a new common framework. First step of such study is to analyse their GL models in order to define their similarities and their differences. Moreover, analysing tools and facilities they offer, we can understand what the new common framework can provided to the users; in particular through this analyses, we can understand if the tools provided by the two approaches are or not complementary and thus the potentiality of the new framework.

The paper is organized as follows. In Section 2 we present the comparison between the GLARE and GRPOVE GL models using the methodology presented in [4]; in Section 3 we extend the analysis taking into account also the tools that GLARE and GPROVE provide to support use and verification of GLs. Finally in Section 4 we address conclusions and future works.

2 Analysis of GLARE and GOSPEL GL Models

Peleg et al. [4] identified eight dimensions to compare different GL approaches. These dimensions regard two broad categories: structuring guidelines as plans of decisions and actions, and linking a guideline to patient data and medical concepts.

We compare GLARE and GPROVE/GOSpeL¹ using these eight dimensions, following the same methodology described in [4]. In order to ground the analysis on a concrete setting, we started our comparison by acquiring the GL for managing chronic cough developed by American College of Chest Physicians [21], that is a revised and updated version of the GL [22] used in [4]. In Figure 1 and Figure 2 we show how parts such GL have been modelled by using GLARE and GOSpeL.

We only provide a short description of the eight dimensions used in [4]; the interested reader is referred to [4] for a detailed discussion about them.

Dimension 1. Organization of guideline plan components: this dimension deals with how the system supports the decomposition of GLs into networks of component tasks (i.e. plans) and how it allows to express various arrangements of these components and their interrelationships.

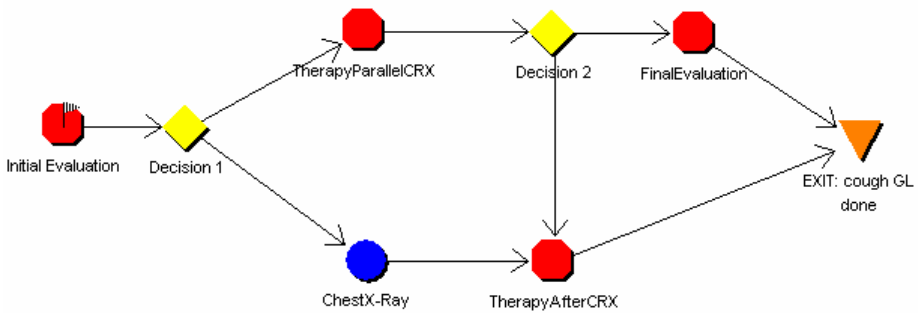


Fig. 1. Part of the chronic cough treatment guideline acquired in GLARE

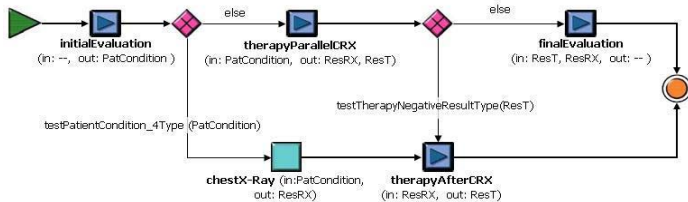


Fig. 2. Part of the chronic cough treatment guideline acquired in GPROVE

GLARE uses the single, generic construct “has-part” to define plans. Plans can be defined in terms of components (see dimension 3 for components types), which could be themselves plans. GLARE supports sequential, parallel, cyclical, and iterative plans. Parallelism is supported through the concurrency relation. Plan iteration can be specified by providing temporal constrains (e.g. maximum and minimum duration, frequency, periodicity, number of repetitions) and/or exit conditions. Although GLARE allows to define only one entry point for a GL, it is possible to specify multi

¹ GOSpeL is the representation language of the GPROVE framework.

entry point referring to patient states in decision criteria or preconditions that affect the guideline control flow.

GOSpeL has a single, generic construct to define a plan and supports nesting. GOSpeL supports sequential, parallel, cyclical, and iterative plans. To support parallelism GOSpeL provides two specific constructs: parallel fork and parallel join, which allow to define respectively the branching point of the multiple paths and synchronization point of the multiple parallel paths. Then GOSpeL provides two ways to define plan iteration through two constructs: the *for* construct given the number of repetition and the *while* construct given a logical guard that works as an exit condition. Although GOSpeL allows to define only one entry point for a GL, it is possible to specify multi entry point using decisions having criteria referring to patient states.

Dimension 2. Specification of goals/intentions: this dimension deals with how the goals/intentions of actions can be defined.

GLARE specifies goals as text string; the goals are presented to the user-physician during the GL execution.

GOSpeL represents goals formally via SCIFF language [20] and uses these expressions to check compliance of a GL execution.

Dimension 3. Model of guideline actions: this dimension deals with the types and the characteristics of GL actions (i.e. the modeling primitives) used to represent the tasks described in GLs.

GLARE distinguishes between *composite* (see Dimension 1 for details) and *atomic* actions. GLARE provides four different types of atomic actions: *work actions*, *query actions*, *decisions* and *conclusions*. Actions are described in terms of their attributes following in ontology described in a set of dedicated databases. In **GLARE** the medical knowledge is stored in a set of databases; the system interacts with them providing the user-physician with information for the GL description.

In particular to specify action attributes, the user-physician can interact with the Pharmacological DB, that stores a structured list of drugs and their costs, with the Resources DB, that contains the resources available in a given hospital, with the ICD DB, that contains a coding system of diseases provided by the Azienda Ospedaliera, and with the Clinical DB, that provides a “standard” terminology to be used when building a GL. For what concerns temporal constraints, GLARE allows to specify qualitative and quantitative constraints, as well as duration of actions, delay between actions, repeated/iterative events; all types of constraints may be imprecise and/or partially defined (see [23] for details). For what concerns the exchange of information (i.e. see System Actions in [4] for details), GLARE models the patient data query, and during the execution the user-physician can specify the execution failure of one action.

GOSpeL describes a GL using *blocks*. The blocks are grouped into three families: *activities* blocks which represent guideline activities at the desired abstraction level; *gateways* blocks used to manage the convergence and the divergence of control flow; *start* and *end* blocks used to represent start and end points of (sub)processes. Activities can be *complex* (see Dimension 1 for details) or *atomic*. Atomic activities model a single atomic working step within the GL (i.e. a situation where a guideline participant should perform something). GOSpeL adopts an ontology-based approach to represent domain-related knowledge and its ontology is defined by two taxonomies:

one called *Activities*, which models activities at the desired abstraction level and divides them in administrative and clinical activities; a second taxonomy called *Entities*, which is used to describe domain's entities characterizing the guideline. Example of *Entities* mapped as *notLivingActor* are the healthcare structures, e.g., the general medicine, radiology and gastroenterology departments, and the different types of analysis results involved in the guideline. Each atomic activity block is semantically specified by mapping it onto an ontological activity and a set of participants. Temporal relations between blocks are expressed using *relations*, which represent causal binary connections between blocks and show how the flow navigates through blocks, imposing a partial ordering among them. Moreover in GOSpeL the temporal constraints between blocks can be defined as CLP constraints [24], allowing to express qualitative and quantitative constraints, as well as duration of actions, delay between actions, repeated/periodic events; all types of constraints may be imprecise and/or partially defined.

Dimension 4. Decision models: this dimension deals with the modeling methodologies for supporting decision making.

GLARE allows to represent two different kinds of decisions: *therapeutic* and *diagnostic* decisions. These decisions are not automatic, and thus the user-physician must always make her/his choice between alternative paths, since GLARE only shows information about whether a path is supported or not. In GLARE the criteria of a *diagnostic decision* are defined by a set of triples <diagnosis, parameter, score> (where a parameter is a triple <data, attribute, value>), plus a threshold to be compared with the different diagnoses' scores. Instead, in *therapeutic decisions* the decision is based on a pre-defined set of qualitative parameters: effectiveness, cost, side-effects, compliance, duration. Observe that in kinds of decision it is possible to express preferences for one or more alternative paths, that are presented in the guideline. Moreover, to support decision making, GLARE provides the decision theory facility (see Section 3 and [25] for more details). Observe that even if GLARE has not a switch construct, in which a decision is taken in a deterministic way, there are some types of automatic decision: the decision concerning the execution of an action with preconditions and of a cyclic action with exit conditions is taken deterministically by the system.

GOSpeL allows to define decisions using a *switch* construct called *exclusive choice*, in which the criteria are logical guard, where each guard is associated with an outgoing path and paths, mutually exclusive, are used to represent at design time all the expected alternative decision derivations. Moreover, GOSpeL provides another kind of decision, called *deferred choice*, which models the decision autonomously taken by a participant (e.g. execute or not the PAP-test); this decision is nondeterministic and is not associated to explicit conditions on the alternative paths. Note that in GOSpeL the user-physician can not specify preferences for alternative paths with respect to a nondeterministic decision.

Dimension 5. Expression/criterion languages used to specify decision criteria: this dimension deals with the languages used to represent decision criteria, including

pre- and post-conditions of GL plan components, and criteria that control plan execution states.

GLARE allows to express presence criteria (criteria concerning giving explicit definitions of terms to be checked). As described above, GLARE uses a threshold policy for what concerns diagnostic decision. Pre-conditions can regard the patient data (in this case they are defined as diagnostic decision criteria) and/or the presence of resources. Exit conditions are based on patient data (defined as diagnostic decision criteria), and the user-physician define whether all or at least n of them should be satisfied (where n is defined during the acquisition phase) to evaluate whether the exit condition of a cyclic action is satisfied during the execution. In GLARE the user-physician can not define temporal criteria concerning the time stamp of patient data and context-dependent expressions. However during the execution of GL the user-physician can decide whether a patient datum is reliable or not. Moreover GLARE does not provide *if...then...else* and *switch* statements in its expression language. Observe that GLARE provides templates to support the criteria definition.

GOSpeL models presence criteria by defining a Boolean data item on the data of the patient. This item is treated like all other data items and can be used in the decisional criteria. GOSpeL does not directly support pre- and post-conditions. They can be modelled using *exclusive choice* to tests conditions and to control the flow accordingly. Moreover, in GOSpeL it is possible to define criteria concerning the time stamp of patient data and context-dependent expressions. *If...then...else* statements can be modeled with *exclusive choice* as well: one outgoing connection of the *exclusive choice* construct is always associated with the *else* label, chosen if none of the conditions attached to the other connections turns out to be *true*.

Dimension 6. Data interpretations/abstractions: this dimension deals with the presence and the characteristic of abstractions, which aid in conceptualizing guideline logic and interpreting data. [4] identifies four types of abstractions: temporal abstractions/temporal patterns (trends), definitions of abstract terms, terminology abstractions via classification hierarchies, and scenarios and patient-state steps (discussed above in Dimension 1).

GLARE does not support temporal abstractions to abstract conditions that persist over time, based on raw, time-stamped values; it is possible to define new abstract terms, but these definitions are not based on formal expressions regarding patient data and/or other concepts (e.g. the user-physician can define the new *isolated systolic hypertension* concept, but s/he can not define its semantics as constrained by systolic blood pressures of at least 140 mmHg, by diastolic blood pressures less than 90 mmHg, and by the situation in which patients are not taking anti-hypertensive drugs). In GLARE the medical knowledge stored in a set of databases (see Dimension 3) follows a taxonomy-based organization via classification hierarchies.

GOSpeL does not support temporal abstractions too. The medical knowledge is organized in an ontology (see Dimension 3) formed by two taxonomies built using the Protégé tool [26]. The users can define new abstract terms, which can also be based on formal expressions regarding patient data and/or other concepts (e.g. the user-physician can define the new *isolated systolic hypertension* concept, and can define its semantics as constrained by systolic blood pressures of at least 140 mmHg, by diastolic blood pressures less than 90 mmHg, and by the situation in which patients are not taking anti-hypertensive drugs).

Dimension 7. Representation of a medical concept model and its use: this dimension deals with how medical concept can be represented in a model and then used.

In **GLARE** the medical knowledge is stored in a set of databases (see Dimension 3); the system interacts with them providing the user-physician with a “standard” vocabulary.

GOSpeL adopts an ontology-based approach to represent clinical knowledge; the user-physician can interact with two taxonomies: *Activities* and *Entities* (see the description in Dimension 3).

Dimension 8. Patient information model: this dimension deals with how the patient information model is defined. This model concerns also the definition of terminologies and of the structure of patient data.

In **GLARE** the patient data model is defined in the Clinical DB (see Dimension 3), which provides a “standard” terminology, and stores the descriptions and the set of possible values of clinical data. The patient data are stored in a specific Patient DB.

In **GOSpeL** a taxonomy is used (see Dimension 3) to describe domain’s entities, namely actors, objects and terms. Every element is defined and managed through the Protégé tool [26]. A specific module called *Event Mapping* can provide the connection with the patient database of the different healthcare organizations.

3 Analysis of Tools Provided by GLARE and by GPROVE

Obviously, the GL model is an important feature, but it is not the only one which can be investigated to characterize a GL computer-assisted manager. We believe it is important to analyze also the GL computer-assisted managers from the point of view of the tools provided, to support the user-physician at differing stages within the GL life-cycle. In this section, we show the results of our analysis concerning tools and facilities provided by GPROVE and by GLARE.

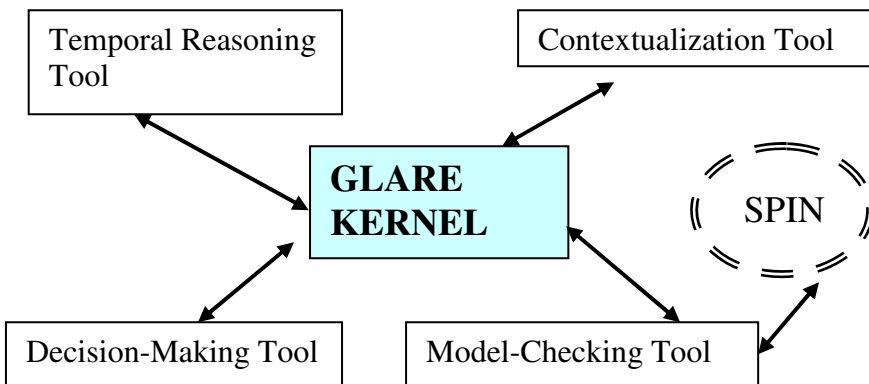


Fig. 3. The GLARE architecture

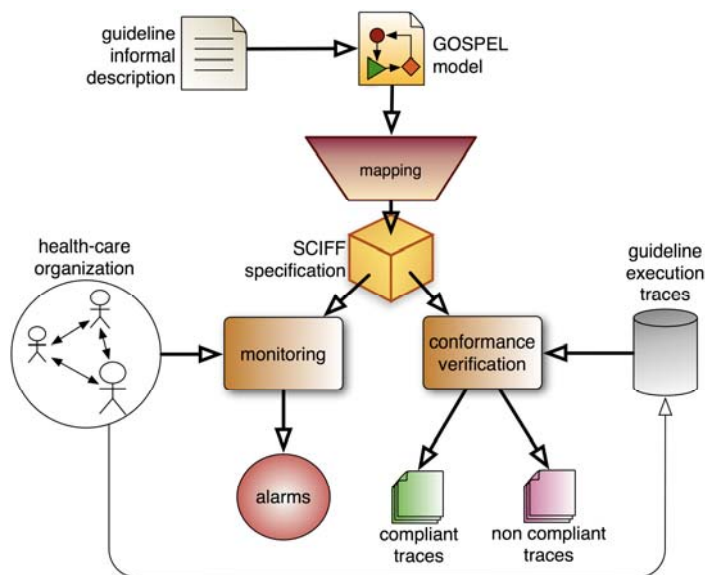


Fig. 4. The GPROVE framework

A first important difference between the GLARE and the GPROVE approaches regards their goals. The GPROVE (its architecture is shown in Figure 4), framework focuses on the compliance verification of the process executions with respect to the specified models, whereas GLARE (its architecture is shown in Figure 3) mainly focuses on GLs acquisition and execution. Of course, such a different focus is reflected in the different tools that they embed.

Both approaches provide a graphical editor that allows user-physicians to acquire and model GL easily. GPROVE proposes the GOSpeL Editor. In GOSpeL, a GL is composed by two parts: a flow chart, which models the process evolution through a graphical language, and an ontology, which describes at a fixed level of abstraction the application domain and gives a semantics to the diagram. The ontology can be developed by using an ontology editor such as Protégé [26]. In GLARE, the graphical editor allows to acquire a GL by means of primitives for drawing the control information within the GL and ad hoc windows to acquire the internal properties of the objects. Furthermore, during GL acquisition GLARE provides a set of facilities to check consistency and terminological correctness. For what concerns temporal constraints, GLARE provides a high-level language to easily represent the temporal action aspects. It then supports the possibility of checking, during acquisition, the temporal consistency of the GL, by exploiting a temporal reasoning tool, which operates in polynomial time on the number of GL actions [23]. In this phase, GLARE also provides a contextualisation tool to manage the gap between the generality of GL themselves (as defined, e.g., by specialists' committees) and the peculiarities of the specific application contexts. This tool allows to adapt GLs on the basis of the resources available in a given context (see [29] for details).

For what concerns compliance, the GPROVE framework is able to check the compliance of a given partial or complete history of a specific execution (i.e., the set of

already happened events recorded in an event log) with the GL via the SOCS-SI tool [27], based on an abductive proof procedure named SCIFF [20]. GLARE supports the check of compliance on temporal constraints only. This check is done by a temporal reasoning tool [23] that evaluates whether the temporal constraints in the GL have been respected or not by the instances of actions that have been executed on the specific patient. Obviously the two approaches are quite different; in particular, the GPROVE approach is more powerful than GLARE in this setting, because it does not only check if temporal constraints are respected, but is able to reason about actions and their data, checking if the behaviour expected by the GL model is actually followed by the concrete participants in a specific case. Furthermore, compliance verification can be seamlessly carried out during the execution, by dynamically acquiring the occurrence of events, or a-posteriori, analyzing already completed executions. For example, in [16] GPROVE has been successfully employed to formalize the process described in the Cervical Cancer Screening Guideline proposed by the sanitary organization of the Emilia Romagna region of Italy [2] and to check the adherence of 1950 concrete screening executions to such process formalization. The results of such analysis were useful to revise the former formalization and identify some relevant characteristics of the screening process.

A variant of the SCIFF proof procedure can be exploited to perform static verification, aiming at identifying design errors and inconsistencies in a GL model. The same verification can be also done in GLARE, which is loosely coupled with the model checker SPIN [28].

Since actually GPROVE framework does not have an execution engine, it does not provide tools and facilities for managing the execution. On other hand, GLARE, which has an execution engine, considers decision making as a crucial issue and provides different facilities to support it. First of all it incorporates a set of facilities based on the temporal reasoning tool: the user-physician can perform queries to obtain temporal information concerning a GL specific execution and can use a simulation facility to see the temporal consequences of choosing among different alternative paths. Moreover via the decision making tool, GLARE provides a decision theory facility, which allows the user-physician to identify the optimal policy, and to calculate the expected utility along a path by exploiting classical powerful dynamic programming algorithms (see [25] for details), and allows also to calculate costs, time and resources required to complete paths in a GL (see [30] for details).

As illustrated above GLARE and GPROVE have different goals and their tools reflected these goals. In particular, the tools embed provided support in different part of lifecycle GL (as show in Table 1). In particular, only GLARE provides a set of facilities to user-physician during acquisition phase, and a Decision-making tool to support user-physician during the execution of a GL on a specific patient. Both GLARE and GPROVE support property verification: GLARE via a model-checking tool based on SPIN and GPROVE via SCIFF proof procedure. Instead only GPROVE provides a set of facilities, which regards the compliance verification of the process executions with respect to the specified models. In particular GPROVE can check the compliance of a given partial or complete history of a specific execution with the GL via the SOCS-SI tool, and provides a monitoring tool, which work on partial GL history. Table 1 proposes a synthesis of the comparison between GLARE and GPROVE facilities and tools.

Table 1. Synthesis of the comparison between GLARE and GPROVE tools

	<i>GLARE</i>	<i>GPROVE</i>
Acquisition	Graphical editor, checking consistency and terminological correctness, checking the temporal consistency, contextualization tool	GOSpeL editor
Execution	Decision-Making Tool	---
Monitoring	---	SCIFF proof procedure on a partial execution
Conformance verification	---	SCIFF proof procedure on a complete or partial execution
Property verification	Model checking tool (via SPIN)	SCIFF proof procedure

4 Conclusions and Future Works

In this work we have compared the GLARE and GPROVE approaches using the methodology proposed in [4], and moreover by taking into account also the tools they provide.

In [4] the authors show that, for what concerns GL model, the six approaches analysed considering eight dimensions have both areas of considerable similarity and areas where different solutions have been adopted to face the same problem.

Thanks to the analysis presented in Section 2, we observe that GLARE and GPROVE propose a solution for almost all the eight dimensions and share with the six approaches analysed in [4] the similar basic features identified as requirements to define a standard for a GL modelling and management system. GLARE and GPROVE GL models have many similarities, especially for what regards the control-flow dimension, even if they slightly differ in the kind of GL knowledge they can represent: GLARE GL model provides a more easy and intuitive way to define the procedural knowledge whereas GPROVE GL models is more oriented to express declarative knowledge and constraints.

On the other hand, our study shows that the provided tools and facilities are quite complementary and focused on different goals (see discussion in section 3). GLARE provides a more rich set of tools than GROVE, mainly because the GLARE project has been started more than ten years ago, while GPROVE has been proposed only recently. In particular, GLARE provides features to support GL acquisition and to support user-physicians in the decision making process during the execution of GL on a specific patient, while GPROVE provides very interesting features for what regards run-time/a-posteriori compliance verification, thanks to the possibility of automatically mapping GOSpeL models to the SCIFF formal framework.

The similarities between their GL models pointed out in this work show the feasibility of integrating the two systems. The idea of common framework is also supported by result of study regarding the tools provided. As a matter of fact, their tools are complementary, and thus such new common framework will encompass all the GLARE facilities and also a powerful compliance verification module. We therefore

consider this work as a first step towards such integration. The second step has been presented at AIME' 09 [31], where we have defined a framework based on the integration between GLARE and GPROVE in order to use SCIFF formal framework to evaluate the GL conformance. The next step will be to investigate how the mapping from GOSpeL to SCIFF could be extended and adapted in the context of GLARE.

Acknowledgments. We would like to thank Manuela Rossi for her contribution to the comparison between GLARE and GPROVE. This work has been partially supported by the FIRB Project TOCALIT.

References

- [1] Gordon, C., Christensen, J.P. (eds.): *Health Telematics for Clinical Guidelines and Protocols*. IOS Press, Amsterdam (1995)
- [2] Fridsma, D.B. (Guest ed.): *Special Issue on Workflow Management and Clinical Guidelines*. *Journal of the American Medical Informatics Association* 22(1), 1–80 (2001)
- [3] ten Teije, A., Miksch, S., Lucas, P. (eds.): *Computer-based medical guidelines and protocols: a primer and current trends*. IOS Press, Amsterdam (2008)
- [4] Peleg, M., Tu, S., Bury, J., Ciccarese, P., Fox, J., Greenes, R.A., Hall, R., Johnson, P.D., Jones, N., Kumar, A., Miksch, S., Quaglini, S., Seyfang, A., Shortliffe, E.H., Stefanelli, M.: *Comparing computer-interpretable guideline models: a case-study approach*. *Journal of the American Medical Informatics Association* 10(1), 52–68 (2003)
- [5] Miksch, S., Shahar, Y., Johnson, P.: *Asbru: a task-specific, intention-based, and time-oriented language for representing skeletal plans*. In: *Proc. 7th Workshop on Knowledge Engineering Methods and Languages*, pp. 9–20 (1997)
- [6] Musen, M.A., Tu, S.W., Das, A.K., Shahar, Y.: *EON: a component-based approach to automation of protocol-directed therapy*. *Journal of the American Medical Informatics Association* 3(6), 367–388 (1996)
- [7] Ohno-Machado, L., Gennari, J.H., Murphy, S., Jain, N.L., Tu, S.W., Oliver, D.E., et al.: *The guideline interchange format: a model for representing guidelines*. *JAMIA* 5(4), 357–372 (1998)
- [8] Peleg, M., Boxawala, A.A., et al.: *GLIF3: The evolution of a guideline representation format*. In: *Proc. AMIA 2000*, pp. 645–649 (2000)
- [9] Quaglini, S., Stefanelli, M., Cavallini, A., Miceli, G., Fassino, C., Mossa, C.: *Guideline-based careflow systems*. *Artificial Intelligence in Medicine* 20(1), 5–22 (2000)
- [10] Johnson, P.D., Tu, S.W., Booth, N., Sugden, B., Purves, I.N.: *Using scenarios in chronic disease management guidelines for primary care*. In: *Proc. AMIA Annu. Fall Symp.*, pp. 389–393 (2000)
- [11] Fox, J., Johns, N., Rahmzadeh, A., Thomson, R.: *Disseminating medical knowledge: the PROforma approach*. *Artificial Intelligence in Medicine* 14, 157–181 (1998)
- [12] Chesani, F., Lamma, E., Mello, P., Montali, M., Storari, S., Baldazzi, P., Manfredi, M.: *Compliance checking of cancer-screening careflows: an approach based on computational logic*. In: ten Teije, A., Miksch, S., Lucas, P. (eds.) *Computer-based medical guidelines and protocols: a primer and current trends*. IOS Press, Amsterdam (2008)
- [13] Isern, D., Moreno, A.: *Distributed guideline-based health care system*. In: *Proceedings of 4th International Conference on Intelligent Systems Design and Applications, ISDA 2004*, pp. 145–150. IEEE Press, Budapest (2004)
- [14] Skonetzki, S., Gausepohl, H.J., van der Haak, M., Knaebel, S., Linderkamp, O., Wetter, T.: *HELEN, a Modular Framework for Representing and Implementing Clinical Practice Guidelines*. *Methods Inf. Med.* 43, 413–426 (2004)

- [15] Dube, K.: A Generic approach to supporting the management of computerised clinical guidelines and protocols, PhD thesis, Institute of Technology, Dublin, Ireland (2004)
- [16] Terenziani, P., Montani, S., Bottrighi, A., Molino, G., Torchio, M.: Applying artificial intelligence to clinical guidelines: the GLARE approach. In: ten Teije, A., Miksch, S., Lucas, P. (eds.) *Computer-based medical guidelines and protocols: a primer and current trends*. IOS Press, Amsterdam (2008)
- [17] De Clercq, P.A., Hasman, A., Blom, J.A., Korsten, H.H.M.: Design and implementation of a framework to support the development of clinical guidelines. *International Journal of Medical Informatics* 64, 285–318 (2001)
- [18] Berg, D., Ram, P., Glasgow, J.: SAGEDesktop: an environment for testing clinical practice guidelines. In: *Proceedings of 26th Annual Conference of the IEEE Engineering in Medicine and Biology Society (IEBMS 2004)*, San Francisco, USA, vol. 2, pp. 3217–3220. IEEE Press, Los Alamitos (2004)
- [19] Chesani, F., De Matteis, P., Mello, P., Montali, M., Storari, S.: A framework for defining and verifying clinical guidelines: A case study on cancer screening. In: Esposito, F., Raś, Z.W., Malerba, D., Semeraro, G. (eds.) *ISMIS 2006. LNCS (LNAI)*, vol. 4203, pp. 338–343. Springer, Heidelberg (2006)
- [20] Alberti, M., Chesani, F., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: Verifiable agent interaction in abductive logic programming: the SCIFF framework. *ACM Transactions on Computational Logic* 9(4), 1–43 (2008)
- [21] Irwin, R.S., Boulet, L.S., Cloutier, M.M., et al.: Managing cough as a defense mechanism and as a symptom: a consensus panel report of the American College of Chest Physicians. *Chest* 114(2), 133–181 (1998)
- [22] Irwin, R.S., Baumann, M.H., Bolser, D.C., Boulet, L.P., Braman, S.S., Brightling, C.E., Brown, K.K., Canning, B.J., Chang, A.B., Dicpinigaitis, P.V., Eccles, R., Brendle Glomb, W., Goldstein, L.B., Graham, L.M., Hargreave, F.E., Kvale, P.A., Zelman Lewis, S., McCool, F.D., McCrory, D.C., Prakash, U.B.S., Pratter, M.R., Rosen, M.J., Schulman, E., Shannon, J.J., Hammond, C.S., Tarlo, S.M.: Diagnosis and management of cough executive summary: ACCP evidence-based clinical practice guidelines. *Chest* 129, 1–23 (2006)
- [23] Anselma, L., Terenziani, P., Montani, S., Bottrighi, A.: Towards a comprehensive treatment of repetitions, periodicity and temporal constraints in clinical guidelines. *Artificial Intelligence in Medicine* 38, 171–195 (2006)
- [24] Jaffar, J., Maher, M.J.: *Constraint logic programming: a survey*. JLP 19-20, 503–582 (1994)
- [25] Montani, S., Terenziani, P.: Exploiting decision theory concepts within clinical guideline systems: towards a general approach. *International Journal of Intelligent System* 21, 585–599 (2006)
- [26] Protégé ontology editor, <http://protege.stanford.edu/>
- [27] The SCIFF abductive proof procedure, <http://lia.deis.unibo.it/Research/sciff/>
- [28] Holzmann, G.J.: *The SPIN Model Checker. Primer and Reference Manual*. Addison-Wesley, Reading (2003)
- [29] Terenziani, P., Montani, S., Bottrighi, A., Torchio, M., Molino, G., Correndo, G.: A context-adaptable approach to clinical guidelines. In: Fieschi, M., et al. (eds.) *Proc. MEDINFO 2004*, pp. 169–173. IOS Press, Amsterdam (2004)
- [30] Terenziani, P., Montani, S., Bottrighi, A., Torchio, M., Molino, G.: Supporting physicians in taking decisions in clinical guidelines: the GLARE “what-if” facility. In: *JAMIA Symposium supplement*, pp. 772–776 (2002)
- [31] Bottrighi, A., Chesani, F., Mello, P., Molino, G., Montali, M., Montani, S., Storari, S., Terenziani, P., Torchio, M.: A Hybrid Approach to Clinical Guideline and to Basic Medical Knowledge Conformance. In: Combi, C., Shahar, Y., Abu-Hanna, A. (eds.) *AIME 2009. LNCS (LNAI)*, vol. 5651, pp. 91–95. Springer, Berlin (2009)