Computational Intelligence, Volume 25, Number 3, 2009

CASE-BASED REASONING FOR MANAGING NONCOMPLIANCE WITH CLINICAL GUIDELINES

Stefania Montani
Università del Piemonte Orientale, Alessandria, Italy

Despite the recognized advantages that can be obtained in clinical practice when following clinical guidelines (GL), situations of noncompliance with them may emerge. Keeping track of such deviations from the default GL execution, and documenting the physician's motivations, would clearly be an added value. Moreover, repeated alterations of GL actions (or flow) may highlight the need for an adaptation of the GL itself to the local reality, or may even indicate an improper or weak initial GL definition.

In this article, we propose an approach for managing noncompliance with GL, based on the case-based reasoning methodology. In front of a new noncompliance case, our tool allows the physician to retrieve past situations similar to the current one, and to decide whether to reapply the same GL modifications adopted in them. Moreover, the tool is able to learn indications from the ground noncompliance cases that can be deployed for local adaptation, and possibly, for suggesting more formal GL revisions to be carried out by a committee of expert physicians.

Key words: case-based reasoning, clinical guidelines, noncompliance, adaptation.

1. INTRODUCTION

Clinical guidelines (GL) can be defined as a means for specifying the "best" clinical procedures and for standardizing them. It has been shown (see, e.g., Overhage et al. 1997) that GL can improve the quality of patient care, reduce variations in quality of care, and reduce costs. Nevertheless, in their conventional, paper-based form, GL often present population-oriented recommendations, and are usually far from being point-of-care facilities (Peleg et al. 2003). Thus, to support physicians in implementing GL into a real hospital setting, a large number of computer-based systems for GL management have been recently proposed in the literature (see, e.g., Musen et al. 1996; Shahar, Miksch, and Johnson 1998; Fox et al. 1998; Shiffman et al. 2000; Peleg and Boxawala 2000; Quaglini et al. 2000; Terenziani, Molino, and Torchio 2001). By offering several functionalities, such as automatic connection to the patient databases, decision-making support, or integration with workflow management systems, such tools can cooperate to make GL really usable and efficient instruments for care provision.

Despite such efforts to facilitate GL adoption and integration in real-world environments, physicians may still decide not to follow the GL indications: while in many nonclinical settings procedures are often very well defined, and strictly dictated by some higher authority, in the medical domain responsibilities are widely shared, and GL users may be *noncompliant* with a GL for a variety of reasons. As a matter of fact, on the one hand a medical professional is aware that biological systems (i.e., patients) are highly unpredictable, and that GL are a set of recommendations which fit an average patient and an average situation (Quaglini et al. 2004), but not necessarily the one under examination. On the other hand, a physician might be scarcely motivated to follow a GL if it is supported by a low level of evidence (see, e.g., Cook et al. 1992). Moreover, the physician herself typically well knows that some biases can affect GL: the knowledge which GL summarize is extracted from clinical trials, but these studies are conducted in more favorable conditions (e.g., with more funds available) with respect to many real-world situations. Finally, she might also have the perception that the GL goal is the one of optimizing resource consumption, and not the one of increasing the quality

Address correspondence to Stefania Montani, Viale Michel 11, I-15100 Alessandria, Italy; e-mail: stefania.montani@unipmn.it

of care (Formoso, Liberati, and Magrini 2001). In summary, noncompliance episodes may emerge due to two categories of reasons: (1) the physician has to face a somehow unpredicted situation, which was not (properly) considered when the GL was issued (in complex, quickly changing domains, it could be possible that a newer version of the GL has addressed such kind of situation, but a (nonspecialized) physician might be unaware of this last version); (2) her professional opinion is different from the one expressed in the GL, and she does not completely trust the GL itself. Keeping track of such deviations from the default GL execution, and documenting the physician's motivations, is obviously a requirement, also from a legal point of view. Moreover, due to the possible presence of biases or to a low level of supporting evidence in a GL, it can be important to evaluate the GL itself on the field, and noncompliance episodes may be exploited for revisions and updates in case of need (Bates et al. 2003). Last, but not least, noncompliance may emerge due to (3) *adaptation* issues. It has been observed that GL dissemination and integration into clinical practice should recognize the multiplicity of working settings within which the GL themselves are meant to be implemented (Scott-Wright et al. 2002; Burgers et al. 2004). A GL developed at the (inter)national level by a group of experts may need to be modified, to reflect the local situation. In particular, two types of adaptation should be considered: (i) local adaptation, that is, adaptation to local constraints in local settings (e.g., hospital resources availability, available practitioners' skills), and (ii) cultural adaptation, that is, the adaptation related to the fact that different countries and/or cultural settings may have different degrees of acceptance of specific clinical procedures, and/or local best practices. Legitimate changes, able to deal with local/cultural constraints, can be made in recommendations, even when the evidence they are based on is the same (Burgers et al. 2004). GL adaptation is so critical that possible methodologies are being studied by international working groups (see, e.g., Fervers et al. 2006) to this hand. However, little formal evaluation has been performed so far; therefore, the identification of proper ways of supporting adaptation is still an open issue, and unsolved adaptation requirements may still generate noncompliance episodes.

In this article, we propose an approach to manage noncompliances with GL, based on the case-based reasoning (CBR) methodology (Kolodner 1993; Aamodt and Plaza 1994). The strength of CBR consists in the significant reduction in the knowledge acquisition efforts, with respect to many other reasoning methodologies. As a matter of fact, acquiring knowledge simply means maintaining the new data, which informally embed a new $\langle situation, action\rangle$ pattern reusable in the future; the system competence is automatically increased as soon as new cases are stored. In particular, an explicit model does not need to be learnt from the data themselves, as it happens in, for example, many machine learning paradigms.

In front of a new noncompliance situation, our tool allows the physician to retrieve past noncompliance cases similar to the current one, and to decide whether to reapply the same GL modifications (i.e., case solutions) adopted in them. Moreover, the tool is able to extract more general indications from the ground cases that can be deployed to suggest GL adaptations, useful to fit the local/cultural implementation context. Additionally, if a repeated alteration of the GL actions (or flow) is adopted by different physicians in front of similar motivations, it may be interpreted as an indicator of an improper or weak GL definition (e.g., due to the presence of biases, changes in evidence, obsolescence of data and/or procedures, etc., Shekelle et al. 2001). If this kind of situation emerges, the tool can be used as a starting point for suggesting a formal GL revision to a committee of expert physicians.[1] Consider

[1] Obviously, the revised version of the GL, if issued, will always be approved by a charged experts team. The tool's suggestions—always checked by the user physician—will be directly adopted only in single noncompliance cases, which would not follow the default GL procedure anyway.

that local adaptation itself may sometimes motivate de novo development of a GL (Fervers et al. 2006).

The article is organized as follows. In Section 2, we describe related work. Section 3 introduces the details of our contribution, while Section 4 is devoted to conclusions and future work.

2. RELATED WORK

Noncompliance with GL, and more generally exceptions to the execution of medical workflow systems, are treated in GUIDE (Quaglini et al. 2001). The system allows a user to redirect, delay, or be noncompliant with a GL action. Noncompliance may lead to the execution of a different action, still chosen among the ones described in the GL (e.g., by altering the control flow), or to the definition of a new action: in this case, the operator has to select it from the SNOMED taxonomy. The requirement is that the *goal* of the GL is still reached, even if by means of a different procedure. Goals allow to verify whether the physician is still compliant with the GL at a high level (i.e., she still shares the GL intentions), even when an exception is generated at the low level (because she has changed some actions or execution flows). Goals as well as exceptions are represented also in GLIF3 (Peleg and Boxawala 2000), in the form of text strings. Here no automation is provided, because the objective is just the one of showing this information to users or of indexing libraries of GL. A more formal description of goals is provided in EON (Musen et al. 1996), PROforma (Fox et al. 1998), and Asbru (Shahar et al. 1998; Duftschmid and Miksch 2001). In Asbru, in particular, any GL modification is released only after its high-level compliance with the GL intentions has been verified by means of a rule-based system. Asbru verification facility also allows to discover anomalies (such as unsatisfiable conditions) that were originally introduced in the GL during the acquisition phase. Other works address the issue of GL verification, for example, by means of model checking (Baumler et al. 2006; Giordano et al. 2006) or theorem proving (tenTeije et al. 2006) techniques. However, the aim of these last works is more the one of discovering logical inconsistencies in the GL or to prove particular properties it exhibits, while we are basically interested in keeping track of noncompliances (typically to a well-formed GL) due to reasons of types (1), (2), and (3) presented in Section 1.

Our CBR-based management of noncompliance is more “lazy” with respect to the discussed literature approaches. We do not aim at checking on-the-fly the adherence of a modification to the original GL intentions, thus we do not model goals or verification rules, whose elicitation might be extremely hard and time consuming. We simply support the physician by showing her past nonformalized examples that match the current *context* (see Section 3.2), and leave to her the responsibility of the final decision. Keeping track of such cases requires a very limited knowledge acquisition effort (because we just store the instances of GL as they were edited and executed in the past), and a limited memory occupancy (because noncompliances are expected to be relatively infrequent). Then, when some cases related to the same context have been acquired, we try to learn some more general suggestions from them, thus preparing a structured proposal for GL modification supported by a set of concrete implementations. Basically, we aim at supporting GL modification (i.e., GL *adaptation* or *revision*) in case of repeated exceptions, more than at verifying the adherence of a single exception to the GL intentions.

As regards local/cultural adaptation support, our work is inserted within an active research panorama, in which various alternative ways of dealing with this issue have been proposed. For instance, Terenziani (Terenziani et al. 2004) proposes to extend the GL representation formalism to take into account the resource requirements associated with each action (and therefore, to each alternative path in the GL): only those actions (paths) whose resource requirements can be satisfied in the given setting (e.g., hospital) can be executed. The approach provides an algorithm that takes in input a "general" GL, and a list of all the locally available resources, and gives in output a new GL, in which all the paths containing nonlocally executable actions have been pruned away. Peleg (2006) deals with the double problem of local adaptation and of a complete integration with the Electronic Patient Record (EPR), and tests the approach on a real-world GL. The authors observe that the two issues at hand have significant affects on the encoding of the GL, including change of algorithm design, definition of decision criteria, and specification of data items that are referenced by the decision criteria themselves; therefore, GL adaptation and integration with the EPR should be considered as early as possible. Scott-Wright et al. (2002) suggest an approach in which the (resource-based) dependencies between actions in a GL can be explicitly described, and where users' modifications to a general GL must respect these dependencies. Finally, also the works examined in the beginning of this section, aiming at preserving the GL goals, can be seen as ways to ensure the adaptability of the procedures to different local or cultural needs, still preserving the GL intentional objectives. However, all of these approaches to GL adaptation appear again to be more knowledge intensive with respect to ours, and therefore more expensive and time consuming if compared to a CBR-based solution.

In the literature, CBR has often been resorted to maintain knowledge about exceptional situations (see, e.g., Marling and Whitehouse 2001; Surma and Vanhoof 1995; d'Aquin, Lieber, and Napoli 2006; Xu 1996 as applications in the medical domain). In particular, in d'Aquin et al. (2006), a CBR approach has been devised to support the physician in changing a specific therapeutic protocol to conform an unusual situation (we could say, in defining a noncompliant version of the protocol; the protocols at hand can be seen as skeletal GL). Nevertheless, only temporary variations of the protocol, for a particular patient, are considered. Long-term protocol changes, obtained from frequently performed modifications, are not dealt with. In Bichindaritz, Kansu, and Sullivan (1998) the authors work on the complementation between theoretical knowledge of GL and experimental knowledge of cases, although here clinical cases illustrate different interpretations of a GL, rather than noncompliance episodes. Finally, CBR has also been used to support process modification in Process Aware Information Systems (PAIS) (Weber, Reichert, and Wild 2006), which must be able to flexibly adapt to (unexpected) process changes as well as to new laws or to reengineering efforts within a business organization. Despite the fact that GL model processes as well, with respect to medical applications PAIS exhibits different needs. In particular, in PAIS model changes are much more frequent. The main concern of the work in Weber et al. (2006) is, thus the one of addressing proper case base maintenance techniques. On the other hand, how to identify redundancies and inconsistencies in the case base (e.g., by evaluating cases that have the same semantic meaning, or by extracting generalizations when possible) is described as a challenge for the future. Thus, the issue of reorganizing the case base to learn well-defined suggestions for supporting long-term GL adaptation or revision, which is a significant need (Scott-Wright et al. 2002; Bates et al. 2003) in practice, to our knowledge has never been systematically treated in the literature, and appears to be a relevant and original contribution of our approach.

3. CBR FOR MANAGING NONCOMPLIANCE AND FOR SUPPORTING GL MODIFICATION

Our approach is currently being implemented within GLARE (Terenziani et al. 2001), a system for acquiring, representing, and executing GL, developed at the Computer Science Department of our university. Nevertheless, our proposal to manage noncompliance and to support GL modification is general enough to be implemented within almost any of the systems for the computerized treatment of GL described in the literature, because it requires the adoption of knowledge representation choices that are widely shared. In particular, basic GL representation primitives and control flow rules are very similar in all the systems; moreover, most of the systems maintain both a general GL description (generated at acquisition time), and a specific GL instance (generated at execution time) that corresponds to the application of the general GL to a specific patient. Details about how these aspects have been afforded in our approach will be provided and resorted to in the remaining of the section, which is organized as follows: Section 3.1 introduces GL representation primitives, Section 3.2 describes case representation, and Section 3.3 deals with case retrieval. Section 3.4 is devoted to illustrate case-based support for GL modification. Finally, Section 3.5 is about testing results.

3.1. GL Representation

In our approach, a GL can be represented as a hierarchical graph, where nodes are the actions to be executed, and edges are the control relations linking them. We can distinguish between atomic and composite actions (plans), where atomic actions represent simple steps in a GL, and plans represent actions that can be defined in terms of their components via the *has-part* relation. The GL itself is a plan. Three different types of atomic actions can be identified: (1) *work actions*, that is, actions that describe a procedure which must be executed at a given point of the GL (e.g., to provide a drug); (2) *decision actions*, used to model the selection among different alternatives (e.g., to decide between surgical and pharmacological treatment for a certain disease); (3) *query actions*, that is, requests of information (typically patient's parameters) that can be obtained from the outside world (physicians, databases, patient's visits or interviews: for example, to ask the patient about her family anamnesis). Actions in a GL are connected through control relations. Control relations establish which actions can be executed next, and in what order. In particular, the *sequence* relation explicitly establishes what is the following action to be executed; the *alternative* relation describes which alternative paths stem from a decision action, and the *repetition* relation states that an action has to be repeated several times.

These primitives will be referred to in the examples of the following sections; in the figures, in particular, boxes will be used to represent work actions, and diamonds to represent decision actions. Control flows will be straightforwardly depicted by arrows.

3.2. Case Representation

In a CBR system, a case usually consists of the following information:

(i) the *problem description* that is, a collection of $\langle feature, value\rangle$ pairs able to summarize the problem at hand;

(ii) the *case solution*, describing the solution adopted for solving the corresponding problem.

In our domain, the *problem description* has to keep track of the *context* that motivated the choice of being noncompliant. The unavailability of a relevant resource or of some needed patient data, or the presence of a patient datum, which is outside the range foreseen by the GL, or very close to a decision threshold defined in the GL, may motivate a noncompliance. Therefore, in our approach the problem description includes patient features (collected by means of query actions in the GL) and local resources information (collected by query actions or modeled as preconditions for executing the GL action at hand). By local resources we mean technological devices and drugs, as well as human competences. To support standardization, the terminology used for features names, and the range of feature values, need to be maintained, for example, as an ontology, to be referenced by all physicians applying the GL itself; operatively, in our approach terms and values are currently stored in proper databases, used as a thesaurus for GL acquisition and execution (Terenziani et al. 2001).

Because a physician may want to adapt/change a GL also when its applicability conditions are met (and thus when the context alone is not sufficient to clarify the reasons for rejecting the GL advice), we also add a free text field (see also Weber et al. 2006) in which she can justify the reasons for her deviation. Such text can provide an insight of the physician's motivations to a colleague that will retrieve that noncompliance case in the future, but will not be resorted to when automatically examining cases for supporting GL modification (see Section 3.4).

Another special feature we introduce is *reputation* (Weber et al. 2006). Reputation is a sort of score, set to 1 when the case is generated, and increased by 1 every time a physician retrieves the case and judges it to be useful for her current problem. Reputation is decreased if the physician retrieves the case, but then discards it. A high reputation is, therefore, an indicator of semantic correctness. Physicians are also encouraged to decrease a case reputation if the modifications suggested in that case resulted in problems when applied in practice.

Finally, we introduce an *intra-case relation* feature to keep track of semantic similarity among cases. The value type of this feature is a case list. How we use it will be explained in Section 3.4.

The *case solution*, on the other hand, is the GL instance properly edited and executed in the situation summarized by the problem description.

3.3. Case Retrieval

When a physician is executing a GL, and does not agree with the GL indications, she may ask our tool to retrieve suggestions on how to restructure the GL itself, held as past cases in the noncompliance case base. The current patient characteristics and resources information represent the context to be used to index the base, and are interpreted as the input case features.

To retrieve the most similar (i.e., less distant) cases with respect to the input one, a measure of distance in the features space has to be provided. Generally speaking, the distance $d(c_i, c_j)$ between cases c_i and c_j can be computed as a weighted average of the normalized distances between their various features, that is

$$d(c_i, c_j) = \frac{\sum_{f=1}^{N} w_f \cdot d(c_i(f), c_j(f))}{\sum_{f=1}^{N} w_f}, \tag{1}$$

where $d(c_i(f), c_j(f))$, and w_f denote the normalized distance between feature f of cases c_i and c_j, and the weight associated with this feature, respectively. Weights can be properly set to state that some features are more "important" for retrieval relatively to the others. They are experimentally set and tuned, and their choice varies from domain to domain.

Various metrics can be relied upon to calculate $d(c_i(f), c_j(f))$; we are currently choosing the heterogeneous Euclidean-overlap metric (HEOM) (Wilson and Martinez 1997), a distance metric able to treat both symbolic and numeric variables, and to cope with the problem of missing data.

In our approach, the solutions of the retrieved cases are then shown to the physician for a personal interpretation. As in Weber et al. (2006), we leave to the user the responsibility of the final decision to be taken in the situation at hand.

3.4. Case-Based Support for GL Modification

After some noncompliance cases have been stored in the case base, we search for redundancies or inconsistencies among them, and we try to identify suggestions for GL modification that, supported by a set of consistent examples, could be taken into consideration for local/cultural adaptation, or—in case of need—for a real GL revision by a committee of experts.

In particular, we aim at learning a more abstract level knowledge from available data, by discovering

- frequent modifications, confirmed by several examples;
- atomic modifications, identified by separating heterogeneous information within a single case;
- semantic relations between different situations.[2]

Note that our strategy for extracting high-level knowledge from ground cases does not require an explicit formalization of the domain knowledge, and a consequent involvement of a domain expert.

What we do is *learning by reorganizing* the case base content, by means of an automatic pair-to-pair comparison between the available ground cases. The comparison between pairs of cases was adopted with the aim of learning adaptation rules/knowledge, for example, in Craw, Wiratunga, and Rowe (2006) and d'Aquin et al. (2007), as well as in earlier contributions. In this work, on the other hand, to goal is not the one of learning adaptation knowledge, because the choice of whether and how to reuse/revise past noncompliance cases to support a current noncompliance case is completely left to physicians. Here, case comparison is a means for maintaining and reorganizing the case base itself. In particular, the identification of frequent changes, and of semantic relations between past noncompliance cases, provides support for a structured long-term GL modification proposal, justified by a set of consistent applied examples.

In this reorganization process, we can distinguish different situations that can emerge when comparing two cases. Such situations can be classified in a clear manner if we examine the process along two *directions*, namely

1. the feature values found in the two case contexts, and

[2] Note that we do not aim at defining a formal mechanism to check semantics, but we just look for similarities in the underlying meaning of the noncompliance episodes we are comparing. The example in Section 3.5 will clarify the point.

2. the modifications proposed in the two case solutions.

We introduce three reorganization *methods* operating on case pairs, whose *declarations* are the following:

- method **merge**
  - input: the two cases to be compared
  - output: one single case that properly merges the input ones
- method **normalize**
  - input: the two cases to be compared
  - output: (up to) three cases, able to separate the heterogeneous information embedded in the input cases
- method **keep separate**
  - input: the two cases to be compared
  - output: the input cases themselves.

The method selection depends on the comparison between the two case solutions (direction 2 above). The case solutions may be completely identical (i.e., they suggest the very same modifications to the GL structure), identical only in some parts (i.e., with a nonempty intersection between the two suggested modification sets), or completely different (i.e., with an empty intersection between the two suggested modification sets). Each one of the three reorganization methods is meant to be applied in exactly one of these situations.

On the other hand, details on each method *implementation* depend on the comparison between the two case contexts (direction 1 above). In particular, the contexts can be completely identical (i.e., the feature lists and respective values are the same in the two case contexts), identical only in some parts (i.e., with a nonempty intersection between the context features sets), or completely different (i.e., with an empty intersection between the context features sets).

By taking into account these two directions, the case base reorganization procedures can be classified as follows:

1. *identical solutions*: method **merge**

   (a) *identical contexts*: this is a redundant situation that can be quickly solved by merging the two instances in just one case. The implementation of **merge** simply keeps one of the two input cases and deletes the other;

   (b) *nonempty intersection between contexts*: the implementation of **merge** here creates a new case, with the same solution as the input ones, and with a context composed by the union of the input contexts;

   (c) *empty intersection between contexts*: we act as above; the rationale of these two last implementations is that we want to merge all the reasons that may have led to a certain GL modification at different times in just one reference case, confirming the suggested changes.

   Figure 1 shows an example, in which case1 and case2 solutions are identical, while contexts are different. The figure abstractly depicts the steps of a GL about the treatment of patients with a bacterial infection. Action C represents the provision of amoxicillin; the other actions are not of interest for the purpose of explaining how the method *merge* works. Both cases are noncompliance examples,

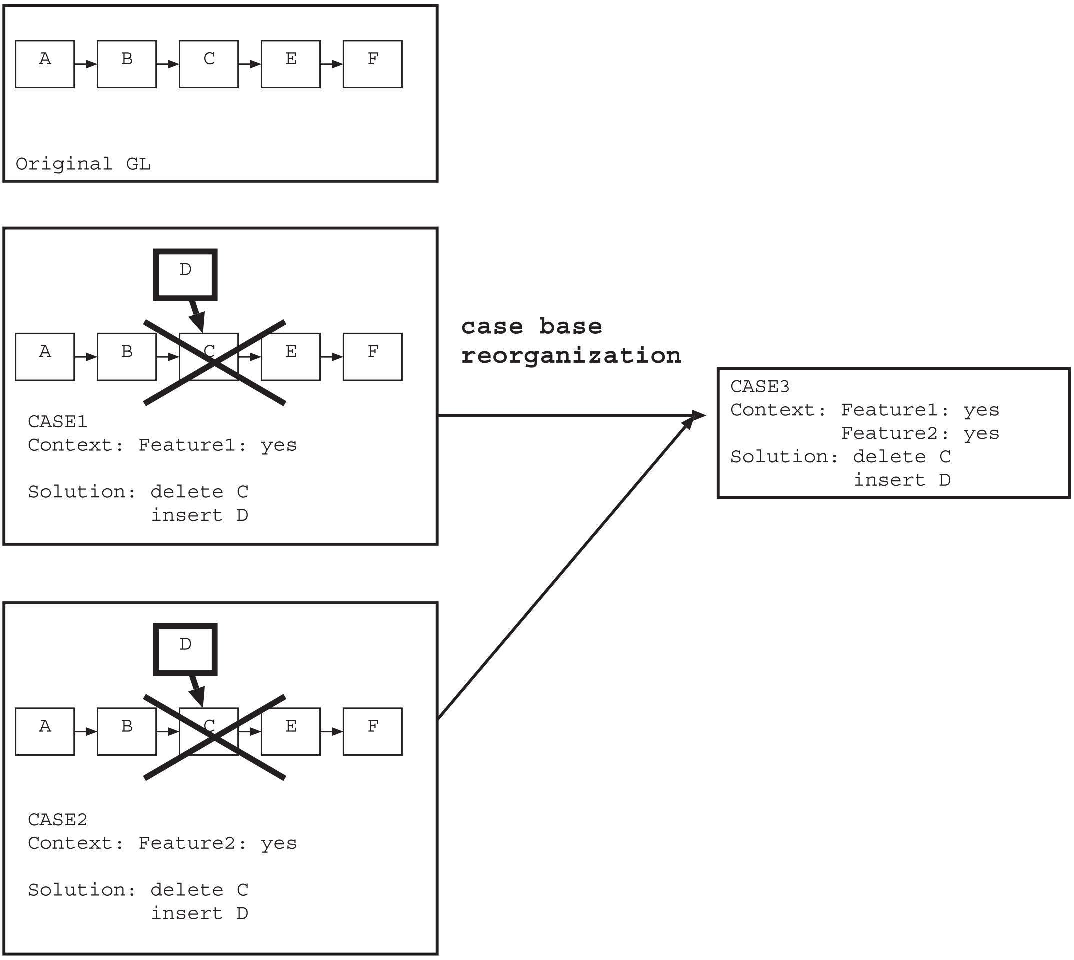


FIGURE 1. Case base reorganization by merging: an example.

because case1 deals with a patient who is allergic to amoxicillin (Feature1), while case2 deals with a patient who has been already cared by providing amoxicillin (which is a moderate-spectrum antibiotic), but unsuccesfully (Feature2): a different class of broad-spectrum antibiotics needs to be adopted for her, to treat the infection. Both cases actually substitute amoxicillin with an antibiotic of a different class (action D). The method *merge* substitues case1 and case2 with a new case (case3), in which the solution (the antibiotic change) is coupled with a context including both allergy and therapy inefficacy. Both original contexts are thus kept, because either the first or the second could justify the therapy change;

2. *nonempty intersection between solutions*: method **normalize**

   (a) *nonempty intersection between contexts*: this situation emerges when heterogeneous information has been stored in the same case. Here, both cases suggest some modifications (the ones in the intersection) due to the common context features.

Moreover, each of the two also embeds additional context information that justify the additional suggested modifications. We propose to afford this situation with a sort of normalization procedure, as the method name suggests. The term is borrowed from relational databases design, where normalization is performed when heterogeneous information is recorded in a single relation, and may lead to anomalies. The implementation of **normalize** derives three cases from the two input ones, as follows: one case maintains the common context/solution pair, thus confirming the shared knowledge, and the other two maintain the additional knowledge implicitly stored in the first and in the second input case respectively. The original cases are deleted.
Note that, if the solution intersection coincides with one of the two solutions, just two cases will be generated.

Figure 2 shows a normalization example, in which case1 presents a solution whose elements are subset of case2 solution. The figure abstractly depicts the steps of a GL about a surgical treatment. In particular action C represents local anesthesia provision, while F represents postsurgery therapy. Both cases are noncompliance examples, because they treat two patients affected by mental deficiency (Feature1) who, in the physician's opinion, would probably not cooperate in the operating theatre. Both cases, thus exclude the local anesthesia option, and substitute it with general anesthesia (action D). Moreover, the patient in case2 suffers from a rare disease (Feature2) and will thus require an additional therapy X. Normalization produces two cases, one obtained by intersecting the context features and keeping the common solution, and one obtained by taking the remaining case2 information;

(b) *identical contexts*: the method **normalize** is implemented similarly to the previous situation, with just one difference; because all features are in common, the three cases will have the same context, and normalization will operate just on solutions. The rationale behind this choice is that we have identified three different groups of modifications (i.e., solution elements), which we want to separate, and that all of them are justified by the same contextual motivations;

(c) *empty intersection between contexts*: we act as aforementioned, but because no common feature can be identified, we build a context made by the union of all the original features in the case maintaining the solution intersection, while we keep unchanged case1 and case2 contexts in the two cases that we separately derive from them. The rationale is that, in this way, in the case storing the common solution part we might come out with a redundant context description but because we do not know what features determined the various modification choices, we are not loosing any potentially useful datum;

3. *empty intersection between solutions*: method **keep separate**

   (a) *empty intersection between contexts*: in this situation the two cases have no relation. Both should be kept as they are. The implementation of **keep separate** leaves the case base as it is;

   (b) *identical contexts*: here, the two cases might share the same semantics, but the GL update is obtained by different procedures. Both cases are kept as they are in the case base, but the method **keep separate** highlights this finding, which may be useful for GL modification. In particular, the method lists the related case identifier in the intra-case relation feature (see Section 3.2) in both the elements of the pair;

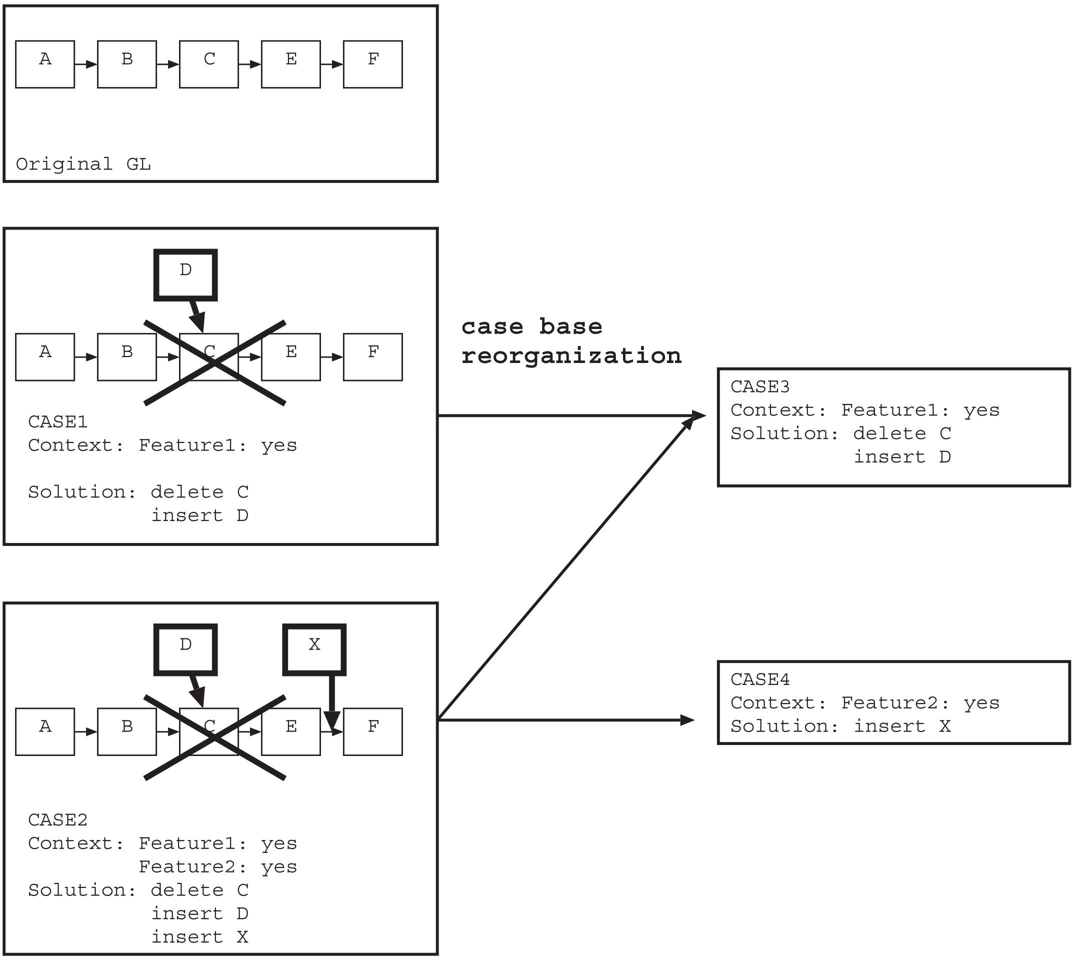


FIGURE 2. Case base reorganization by normalizing: an example.

(c) *nonempty intersection between contexts*: the two cases might share similar semantics, but again the GL update is obtained by different procedures. We act as aforementioned.

Figure 3 summarizes the various alternatives. A set of examples is also provided in Section 3.5.

Note that, each time a case is built by merging or normalizing over two cases, its reputation is calculated as the sum of the input case reputations.

After the case base has been processed by means of this reorganization procedure, the resulting cases, showing a positive reputation, are listed as suggestions for long-term GL modification. A more strict reputation threshold (which appears to be a parameter dependent on the clinical domain in which the GL is meant to be applied) can be manually set by the physician responsible for extracting the suggestions. The physician may also decide to increase reputation by 1 every time the case is linked to another case, by the method **keep separate**.

N.	Solution intersection	Context intersection	Method	Final case context	Final case modifications	Reputation
1(a)	all	all	merge	as case1(2)	as case1(2)	sum
1(b)	all	non empty	merge	union	as case1(2)	sum
1(c)	all	empty	merge	union	as case1(2)	sum

2(a)	non empty	non empty	normalize	intersection
				diff(case1,case2)
				diff(case2,case1)
					intersection
					diff(case1,case2)
					diff(case2,case1)
						sum
						as case1
						as case2

2(b)	non empty	all	normalize	intersection
				intersection
				intersection
					intersection
					diff(case1,case2)
					diff(case2,case1)
						sum
						as case1
						as case2

2(c)	non empty	empty	normalize	union
				as case1
				as case2
					intersection
					diff(case1,case2)
					diff(case2,case1)
						sum
						as case1
						as case2

3(a)	empty	empty	keep separate	as case1 as case2	as case1 as case2	as case1 as case2
3(b)	empty	all	keep separate	as case1 (R) as case2 (R)	as case1 as case2	as case1 as case2
3(c)	empty	non empty	keep separate	as case1 (R) as case2 (R)	as case1 as case2	as case1 as case2

FIGURE 3. Case base reorganization, obtained by comparing two cases (namely, case1 and case2). diff(case1–case2) denotes a set of context features (or elementary modifications) obtained as the difference between the set of context features (or elementary modifications, respectively) in case1 and the ones in case2. An (R) in the final case context column indicates that the intra-case relation feature has been properly set.

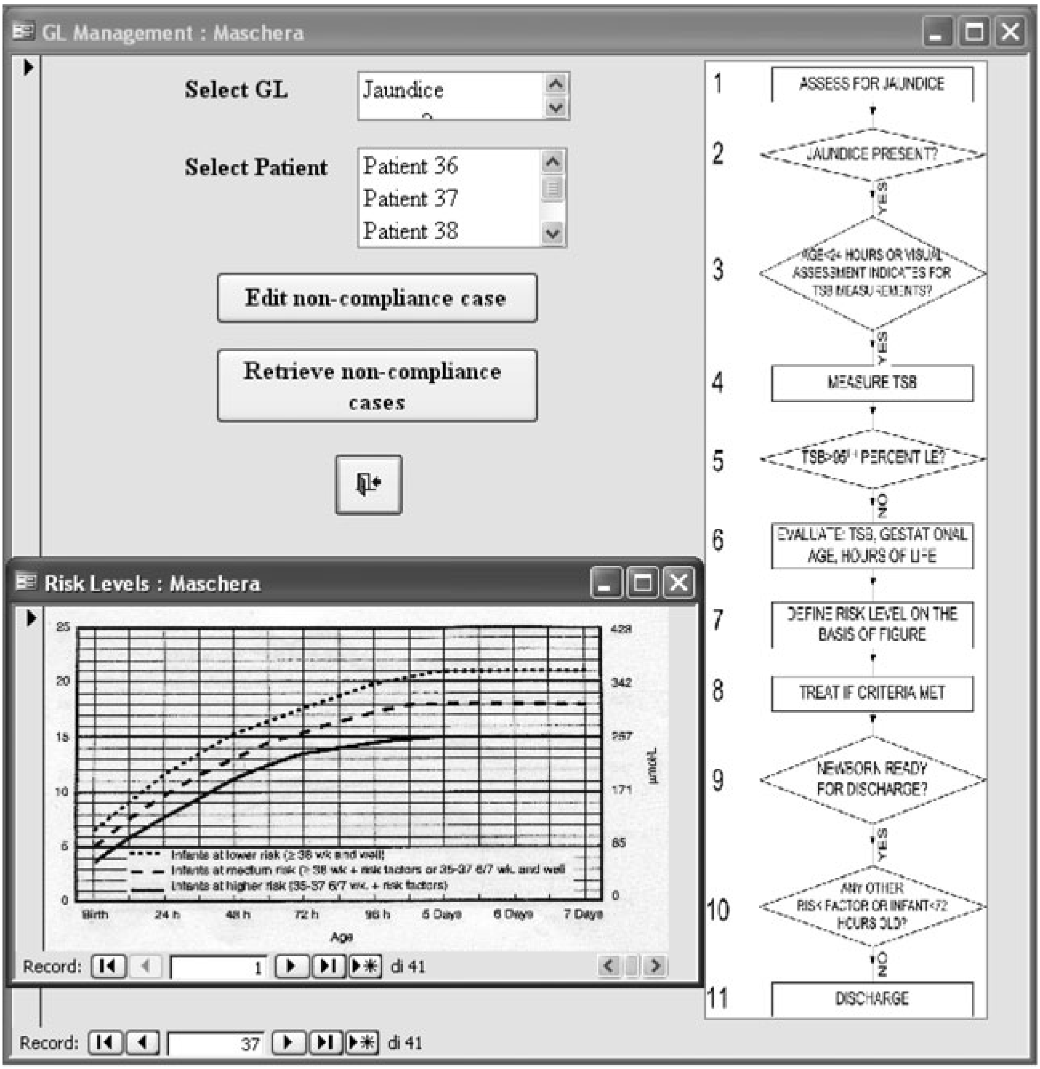
3.5. Testing Results

Since May 2007, we have made the CBR tool described in this article available to the pediatricians of the Obstetrics Department of Policlinico S. Matteo Hospital in Pavia, Italy. The Obstetrics Department is a setting where five to eight newborns are typically admitted every day, and are cared by a team of five physicians. Some testing results are being collected here, and a wider validation is foreseen in the next months.[3]

So far, two relevant successful experiences have been reported by our medical collaborators. They both refer to a jaundice management GL adopted for newborns (American Academy of Pediatrics 2004), which is depicted in Figure 4, as it is represented by means of our tool. In particular, only a subset of the GL actions is shown, for the sake of clarity. A complete version of the GL is available in American Academy of Pediatrics (2004). The GL applies to babies of 35 or more weeks of gestation. The American Academy of Pediatrics clearly states that this GL was based on limited evidence, and might show approximations. The deployment of our tool therefore appears to be particularly suited for this situation, at least for supporting adaptations to the local reality.

Experience 1. The Pavia Obstetrics Department suffers from an insufficient availability of beds for mothers, in front of a growing number of women who choose this site for the birth of their babies. Therefore, if a baby is discharged, but later on she needs to be rehospitalized (e.g., due to jaundice problems), it may become very difficult to readmit the mother as well, thus creating troubles with breast-feeding.

[3] Nevertheless, it is worth noting that the practical reachability of significant validation conclusions appears to be quite hard to be obtained in the short period because noncompliances to a well-formed GL are relatively infrequent.



GL Management : Maschera
Edit non-compliance case
Retrieve non-compliance cases
Risk Levels : Maschera
25
20
15
10
5
0
429
342
257
171
85
0
μmol/L
Birth
24 h
48 h
72 h
96 h
5 Days
6 Days
7 Days
Age
Infants at lower risk (≥ 38 wk and well)
Infants at medium risk (≥ 38 wk + risk factors or 35-37 6/7 wk. and well
Infants at higher risk (35-37 6/7 wk. + risk factors)
1 ASSESS FOR JAUNDICE
2 JAUNDICE PRESENT?
YES
3 AGE<24 HOURS OR VISUAL ASSESSMENT INDICATES FOR TSB MEASUREMENTS?
YES
4 MEASURE TSB
5 TSB>95th PERCENTILE?
NO
6 EVALUATE: TSB, GESTATIONAL AGE, HOURS OF LIFE
7 DEFINE RISK LEVEL ON THE BASIS OF FIGURE
8 TREAT IF CRITERIA MET
9 NEWBORN READY FOR DISCHARGE?
YES
10 ANY OTHER RISK FACTOR OR INFANT<72 HOURS OLD?
NO
11 DISCHARGE

Select GL: Jaundice

Select Patient: Patient 36, Patient 37, Patient 38

Record: 1 di 41

Record: 37 di 41

FIGURE 4. Part of the American Academy of Pediatrics GL for managing jaundice (adapted from American Academy of Pediatrics 2004), visualized by means of our tool. The pop-up window on the left shows the plot about risk levels for phototherapy, referred to in action 7.

By following the American Academy of Pediatrics GL, at 48 h of age total serum bilirubin (TSB) is measured in all babies in whom jaundice by visual assessment appears to be severe enough. Healthy term babies with TSB values between 11 and 13 mg/dl are considered as low-risk subjects; a value between 13 and 15 mg/dl classifies the baby as a medium-risk subject, and a value above 15 mg/dl classifies her/him as a high-risk one. High-risk infants are treated with phototherapy; medium-risk ones are reexamined before discharge (typically at 72 h of age). Thresholds can be lowered for near-term babies and for those with additional risk factors, such as temperature instability, acidosis, etc. (see the pop-up window in Figure 4).

Motivated by beds unavailability in Pavia, one pediatrician generated a noncompliance case (case1), in which a healthy term baby with a TSB = 13.6 mg/dl at 48 h of age was treated with phototherapy. To avoid rehospitalization, the baby was therefore managed as if she was a high-risk infant.

Analogously, a colleague generated a second case (case2) in which a healthy term baby with TSB = 14 mg/dl at 48 h of age was treated with phototherapy as well.

In a third case (case3), the physician suggested to repeat the TSB control before discharge (at 72 h of age), for a healthy term baby with TSB = 12 mg/dl at 48 h of age. The baby was thus treated as if she was a medium-risk one.

The system **merged** case1 and case2 (see Section 3.4), which showed a complete intersection between contexts, due to the identification of the following common (main) features: (i) healthy term newborn, and (ii) medium-risk TSB. The two cases also showed an identical solution, namely: (iii) change of risk level from "medium" to "high" (action 7 in Figure 4); the risk-level upgrade determined the consequent application of phototherapy (action 8). The method **merge** was implemented as in row 1(a) in Figure 3.

The obtained case (case4, with reputation = 2) was later compared with case3, which exhibited the following features in its context: (i) healthy term newborn, and (iv) low-risk TSB. Feature (i) determined a nonempty intersection with case4 context. Case3 solution consisted in the (v) change of risk level from "low" to "medium" (action 7 in Figure 4); the risk-level upgrade this time determined the consequent repetition of TSB measurement/treatment before discharge (action 8). The method **keep separate** was applied here, as in row 3(c) in Figure 3. As a matter of fact, technically case3 and case4 suggested different solutions, but with very similar semantics: in both situations the risk level was *increased*, and this change led to the application of the proper procedures to face jaundice problems.

The team of physicians working with us formally reviewed this result, and finally decided to routinely adopt the revised GL version obtained by generalizing the hints in case3 and case4. Therefore, phototherapy is now applied to all babies with a medium (or high) TSB value at 48 h of age, and an additional TSB control/treatment is performed before discharge in all babies who showed jaundice, even if in a mild form.

Experience 2. Experience 2 started from a couple of relatively unusual situations that took place in Pavia.

In the first case (case1), a baby with a gestational age of 34 weeks + 5 days, but with a birth weight of more than 2,000 g, was managed by following the GL. At 48 h of age his TSB was 13.1 mg/dl (i.e., quite a good value, significantly below 15 mg/dl, the healthy term newborns high-risk threshold—see the pop-up window in Figure 4). Since he was well and had a good birth weight, and because the TSB value was relatively low, he was considered as an infant at medium risk (instead of an infant at high risk, as his gestational age might suggest). Therefore, he was not treated with phototherapy.

In the second case (case2), a 38-week baby born with a weight of 2,300 g had a very significant weight loss at 48 h of age, because breast-feeding was highly recommended to the mother, but nursing was not going well at all. This infant was treated with phototherapy, even if his TSB was 14.2 mg/dl, and thus below 15 mg/dl. Moreover, a bottle-feeding suggestion was added to the GL actions.

Case1 context included as (main) features: (i) gestational age = 34 weeks + 5 days (which made it technically out of the GL scope; but the value was very close to the admissibility threshold); (ii) weight with respect to the gestational age = nonaverage (namely: high); (iii) medium-risk TSB. Case2 context included: (i) gestational age = 38 weeks; (ii) weight with respect to the gestational age = nonaverage (namely: low); (iii) medium-risk TSB; (iv) food intake = insufficient. Case1 solution suggested: (v) evaluation of weight (action 6); (vi) definition of risk considering weight too (action 7). Case2 solution suggested: (v) evaluation of weight (action 6); (vi) definition of risk considering weight too (action 7);

(vii) introduction of action 6 is between action 6 and action 7 (see Figure 4): bottle feeding (see Figure 4).

By applying the method **normalize** as in row 2(a) in Figure 3, the system identified nonaverage weight and medium-risk TSB (see items (ii) and (iii)) in the context intersection, and weight evaluation (see item (v)) as well as the definition of risk on the basis of weight too (see item (vi)) in the solution intersection. The tool thus incorporated these common elements in a new noncompliance case, with reputation = 2.

The operation was again formally analyzed by the pediatricians working with us, who judged it as sound and meaningful. In particular, the system helped them in defining a new version of the GL, in which also weight (in addition to TSB, gestational age at birth and hours of life—see Section 6) is conceived as a very important indicator for determining the risk level of an infant, and then for suggesting therapy. In the new version of the GL, weight evaluation is added to action 6 and risk assessment (action 7) is changed, to take weight into account. The revised procedure is being routinely applied in Pavia.

Observe that, while the GL adaptation in experience 1 was strongly motivated by the local situation, and leads to additional costs which would not be easily accepted in a different context, the modification in experience 2 might have a more general impact on jaundice treatment: the collection of similar results in different hospital settings might also justify the request of a formal revision of the GL in American Academy of Pediatrics (2004), to produce a new GL version suited to properly handle less typical cases like the ones presented in experience 2.

4. CONCLUSIONS AND FUTURE WORK

In this article, we have described an approach for managing noncompliance with GL, and for suggesting GL adaptations or revisions, based on the CBR methodology. With respect to the other literature approaches, we are more interested in supporting long-term GL modification, in case of repeated noncompliance episodes, rather than in verifying the adherence of a single episode to the GL intentions. To this hand, we have defined an automatic strategy to reorganize the case base, to learn high-level knowledge from ground cases.

From a technical viewpoint, several enhancement are foreseen as a future work. First, at the moment we do not take into account the textual motivations introduced by physicians, when we reorganize the case base. An effort to structure this free text feature would be very relevant, because physicians motivations may help in better distinguishing between potentially ambiguous situations, such as the ones in which two cases share (part of) the same solution, but do not share the same features.

Moreover, we could evaluate whether a distinction between action modifications and flow modifications (Ciccarese et al. 2003) could be relevant for reorganizing the case base, thus providing more significant suggestions.

We could also try to automatize the choice of the reputation threshold, which currently has to be manually set.

Because GL are represented as graphs, we will also evaluate the possibility of exploiting learning techniques over graphs to figure out isomorphism, which might be useful to discover common actions that have been taken for different patients (Perner 2003).

In the future, we also plan to complete the integration of our tool as a facility within GLARE (Terenziani et al. 2001). The integration will allow physicians to insert the revised version of the GL by means of the GLARE acquisition interface, to easily build the noncompliance case base. Note that GLARE is also interfaced with the EPR; therefore, all the needed patient data for the GL execution will be automatically retrieved. Then, a more extensive validation will be carried out at Policlinico S. Matteo, where the first testing experiences are showing encouraging results, at least as regards GL local adaptation support.

It is worth noting that our approach explores a research direction, namely, the one of exploiting CBR for clinical knowledge discovery/update, which is recently emerging as a significant trend in the literature. For instance, the idea of using cases to possibly complement the knowledge formalized whithin a GL, or even to generate hypotheses to amend it, is envisioned also in Bichindaritz and Marling (2006). More generally, the work in Bichindaritz (2007) relies upon CBR to mine knowledge from the biomedical literature, by taking advantage from the exploitation of prototypes, which allow to face the otherwise overwhelming rate of knowledge creation in this domain.

It is also worth noting that, despite the fact that this article refers to a medical application domain, the CBR approach presented here could also be seen as a valuable methodology for supporting process update in the PAIS domain (to our knowledge, CBR has not been extensively relied upon to this end, see just Weber et al. 2006). As a matter of fact, we have recently implemented a tool with analogous functionalities with respect to the ones described in this work for Interporto di Rivalta Scrivia, Italy. Interporto is one of the largest logistic centers in northern Italy; at Interporto, several plants for food preprocessing are also installed and routinely used. Engineers and employees responsible for plant administration and security follow specific protocols (similar to GL) for designing and executing plant maintenance procedures. Our tool, which is interfaced with the Interporto DBMS, is enabling them to keep track of protocol alterations (which are relatively frequent), and to learn high-level knowledge from such noncompliance cases, which can be useful to restructure the protocols themselves. The tool is currently being tested, and the first evaluation results from this industrial setting will be soon available. At that time, we will be allowed to draw some more general conclusions about our CBR approach for process adaptation and revision, beyond the specific example of medical processes (i.e., GL).

ACKNOWLEDGMENTS

The author is grateful to the pediatricians of the Obstetrics Department of Policlinico S. Matteo hospital in Pavia, and in particular to Dr. Elisa Mongini, for being the leader of the testing activities, and for her useful comments on the case base reorganization strategy.

REFERENCES

AAMODT, A., and E. PLAZA. 1994. Case-based reasoning: Foundational issues, methodological variations and systems approaches. AI Communications, **7**:39–59.

AMERICAN ACADEMY OF PEDIATRICS. 2004. Clinical practice guideline: Management of hyperbilirubinemia in the newborn infant 35 or more weeks of gestation. Pediatrics, **114**:297–316.

A. SCOTT-WRIGHT, A. A. BOXWALA, Y. DENEKAMP, R. A. GREENES, D. TATE. 2003. Applying axiomatic design methodology for guideline revision. *In* the American Medical Informatics Annual Symposium Proceedings. *Edited by* Mark Musen. AMIA, Bethesda, MD, p. 1000.

BATES, D. W., G. J. KUPERMAN, S. WANG, T. GANDHI, A. KITTLER, L. VOLK, C. SPURR, R. KHORASANI, M. TANASIJEVIC, AND B. MIDDLETON. 2003. Ten commandments for effective clinical decision support: Making the practice of evidence-based medicine a reality. JAMIA, **10**:523–530.

BAUMLER, S., M. BALSER, A. DUNETS, W. REIF, and J. SCHMITT. 2006. A verification of medical guidelines by model checking—A case study. *In* Proceedings of the Model Checking Software SPIN. Vienna.

BICHINDARITZ, I., and C. MARLING. 2006. Case-based reasoning in the health sciences: What's next? Artificial Intelligence in Medicine, **36**:127–135.

BICHINDARITZ, I., E. KANSU, and K. SULLIVAN. 1998. Case-based reasoning in care-partner: Gathering evidence for evidence-based medical practice. *In* Proceedings of the 4th European Workshop on Case-Based Reasoning, **Vol. 1488**. *Edited by* B. Smyth and P. Cunningham. *Lecture Notes in Computer Science*, Springer, Dublin, Ireland, pp. 334–345.

BICHINDARITZ, I. 2007. Prototypical case guided case-based reasoning in biomedical domains. *In* Proceedings of the Case-Based Reasoning in the Health Sciences Workshop, International Conference on Case Based Reasoning (ICCBR), Belfast.

BURGERS, J. S., B. FERVERS, M. HAUGH, M. BROUWERS, G. BROWMAN, T. PHILIP, and F. A. CLUZEAU. 2004. International assessment of the quality of clinical practice guidelines in oncology using the appraisal of guidelines and research and evaluation instrument. Journal of Clinical Oncology, **22**:2000–2007.

CICCARESE, P., E. CAFFI, L. BOIOCCHI, A. HALEVY, S. QUAGLINI, A. KUMAR, M. STEFANELLI. 2003. The newguide project: Guidelines, information sharing and learning from exceptions. *In* Proceedings of the Artificial Intelligence in Medicine Europe (AIME), Springer, Berlin, pp. 163–167.

COOK, D. J., G. H. GUYATT, A. LAUPACIS, and D. L. SACKETT. 1992. Rules of evidence and clinical recommendations on the use of antithrombotic agents. Chest, **102**:305–311.

CRAW, S., N. WIRATUNGA, and R. C. ROWE. 2006. Learning adaptation knowledge to improve case-based reasoning. Artificial Intelligence, **170**:1175–1192.

D'AQUIN, M., J. LIEBER, and A. NAPOLI. 2006. Adaptation knowledge acquisition: A case study for case-based decision support in oncology. Computational Intelligence, **22**:161–176.

D'AQUIN, M., F. BADRA, S. LAFROGNE, J. LIEBER, A. NAPOLI, and L. SZATHMARY. 2007. Case base mining for adaptation knowledge acquisition. *In* Proceedings of IJCAI, Hyderabad.

DUFTSCHMID, G., and S. MIKSCH. 2001. Knowledge-based verification of clinical guidelines by detection of anomalies. Artificial Intelligence in Medicine, **22**:23–41.

FERVERS, B., J. S. BURGERS, M. C. HAUGH, J. LATREILLE, N. MLIKA-CABANNE, L. PAQUET, M. COULOMBE, M. POIRIER, and B. BURNARD. 2006. Adaptation of clinical guidelines: Literature review and proposition for a framework and procedure. International Journal for Quality in Healthcare, **18**:167–176.

FORMOSO, G., A. LIBERATI, and N. MAGRINI. 2001. Practice guidelines: Useful and participative method? Survey of Italian physicians by professional setting. Archives of Internal Medicine, **161**:2037–2042.

FOX, J., N. JOHNS, A. RAHMANZADEH, and R. THOMSON. 1998. Disseminating medical knowledge: The proforma approach. Artificial Intelligence in Medicine, **14**:157–182.

GIORDANO, L., P. TERENZIANI, A. BOTTRIGHI, S. MONTANI, and L. DONZELLA. 2006. Model checking for clinical guidelines: An agent-based approach. *In* Journal of the American Medical Informatics Association (JAMIA) Symposium Supplement.

KOLODNER, J. L. 1993. Case-Based Reasoning. Morgan Kaufmann, San Mateo, CA.

MARLING, C., and D. WHITEHOUSE. 2001. Case-based reasoning in the care of Alzheimer's disease patients. *In* Proceedings of the 4th International Conference on CASE-BASED Reasoning. *Edited by* D. W. Aha and I. Watson. Springer, Berlin, pp. 702–715.

MUSEN, M. A., S. W. TU, A. K. DAS, and Y. SHAHAR. 1996. EON: A component-based approach to automation of protocol-directed therapy. Journal of the American Medical Informatics Association, **6**:367–388.

OVERHAGE, J. M., W. M. TIERNEY, X. H. ZHOU, and C. J. MCDONALD. 1997. A randomized tria; of corollary orders to prevent errors of omission. JAMIA, **4**:364–375.

PELEG, M., and A. A. BOXAWALA. 2000. Glif3: The evolution of a guideline representation format. *In* Proceedings of the AMIA, pp. 645–649.

PELEG, M., S. TU, J. BURY, P. CICCARESE, J. FOX, R. A. GREENES, R. HALL, P. D. JOHNSON, N. JONES, A. KUMAR, S. MIKSCH, S. QUAGLINI, A. SEYFANG, E. H. SHORTLIFFE, AND M. STEFANELLI. 2003. Comparing models of decision and action for guideline-based decision support: A case-study approach. Journal of the American Medical Informatics Association, **10**:52–68.

PELEG, M., D. WANG, A. FODOR, S. KEREN, and E. KARNIELI. 2006. Adaptation of practice guidelines for clinical decision support: A case study of diabetic foot care. *In* Proceedings of the Workshop on AI Techniques in Healthcare: Evidence-Based Guidelines and Protocols (ECAI) Riva del Garda, Italy, August 29, 2006, electronic proceedings, http://mis.hevra.haifa.ac.il/∼morpeleg/pubs/DiabetesGuidelineECAI.pdf, last accessed June 9, 2009.

PERNER, P. 2003. Incremental learning of retrieval knowledge in a case-based reasoning system. *In* Proceedings of ICCBR, LNAI, **Vol. 2689**. *Edited by* K. D. Ashley, D. G. Bridge. Springer-Verlag, Berlin, pp. 422–436.

QUAGLINI, S., M. STEFANELLI, A. CAVALLINI, G. MICELI, C. FASSINO, and C. MOSSA. 2000. Guideline-based careflow systems. Artificial Intelligence in Medicine, **1**:5–22.

QUAGLINI, S., M. STEFANELLI, G. LANZOLA, V. CAPORUSSO, and S. PANZARASA. 2001. Flexible guideline-based patient careflow systems. Artificial Intelligence in Medicine, **22**:65–80.

QUAGLINI, S., P. CICCARESE, G. MICIELI, and A. CAVALLINI. 2004. Non-compliance with guidelines: Motivations and consequences in a case study. *In* Proceedings of the Computer-Based Support for Clinical Guidelines, IOS Press, Amsterdam, pp. 75–87.

SHAHAR, Y., S. MIKSCH, and P. JOHNSON. 1998. The asgaard project: A task-specific framework for the application and critiquing of time-oriented clinical guidelines. Artificial Intelligence in Medicine, **14**:29–51.

SHEKELLE, P., M. P. ECCLES, J. M. GRIMSHAW, and S. H. WOOLF. 2001. When should clinical guidelines be updated? British Medical Journal, **323**:155–157.

SHIFFMAN, R. N., B. T. KARRAS, A. AGRAWAL, L. MENCO R. CHEN, and S. NATH. 2000. Gem: A proposal for a more comprehensive guideline document model using XML. Journal of the American Medical Informatics Association, **5**:488–498.

SURMA, J., and K. VANHOOF. 1995. Integration rules and cases for the classification task. *In* Proceedings of the 1st International Conference on Case-Based Reasoning, **Vol. 1010**. *Edited by* M. Veloso and A. Aamodt. *Lecture Notes in Computer Science* Springer, Sesimbra, Portugal, pp. 325–334.

TEN TEIJE, A., M. MARCOS, M. BALSER, J. VAN CROONENBORG, C. DUELLI, F. VAN HARMELEN, P. LUCAS, S. MIKSCH, W. REIF, K. ROSENBRAND, AND A. SEYFANG. 2006. Improving medical protocols by formal methods. Artificial Intelligence in Medicine, **36**:193–209.

TERENZIANI, P., G. MOLINO, and M. TORCHIO. 2001. A modular approach for representing and executing clinical guidelines. Artificial Intelligence in Medicine, **23**:249–276.

TERENZIANI, P., S. MONTANI, A. BOTTRIGHI, M. TORCHIO, G. MOLINO, and G. CORRENDO. 2004. A context-adaptable approach to clinical guidelines. *In* Proceedings of the World Congress for Health Informatics (MedInfo). *Edited by* M. Fieschi, E. Coiera, and Y. J. Li. IOS Press, Amsterdam, pp. 169–173.

WEBER, B., M. REICHERT, and W. WILD. 2006. Case-based maintenance for ccbr-based process evolution. *In* Proceedings of the European Conference on Case Based Reasoning (ECCBR), LNAI 4106. *Edited by* T. Roth-Berghofer, M. Goker, and H. Altay Guvenir. Springer, Berlin, pp. 106–120.

WILSON, D. R., and T. R. MARTINEZ. 1997. Improved heterogeneous distance functions. Journal of Artificial Intelligence Research, **6**:1–34.

XU, L. D. 1996. An integrated rule- and case-based approach to AIDS initial assessment. International Journal of Biomedical Computing, **40**:197–207.