Author's response to reviews

**Title:** Requirements for Guidelines Systems: Implementation Challenges and Lessons from Existing Software-Engineering Efforts

**Authors:**

Hemant Shah (proteme@gmail.com)
Raymod D Allard (dallard2@hfhs.org)
Robert Enberg (enbergs@gmail.com)
Ganesh Krishnan (gkrishn1@hfhs.org)
Prakash M Nadkarni (prakash.nadkarni@yale.edu)
Patricia Williams (pwilliam@hfhs.org)

**Version:** 2  **Date:** 3 October 2011

**Author's response to reviews:** see over
We thank the reviewers for their feedback. Reviewer 2 appears to have no concerns about the article, and requires no response. Reviewer 1 (Isern) has provided an extremely comprehensive and authoritative review, and the numerous points in the critique are fair and accurate.

In our response below, we reproduce the point made by Reviewer 1 in **bold**, and our response in plain text, and then state how we have revised the text.

**General Concerns:**

**A primary concern was that many of the requirements listed in the paper are not novel. Some of them are widely used in existing computer-based guideline systems, or have been proposed in earlier publications, which were not cited.**

We agree. We have attempted to fix this problem in two ways:

1. Inserting the requisite citations throughout the manuscript.
2. Refocusing the paper on what we consider to be its primary contribution – the emphasis on the challenges of translating ideas embodied in research prototypes into production code. The excellent Isern/Moreno 2008 review points out that none of the clinical guideline systems (CGSs) studied appears to be used in daily patient management. (This has not changed in 2011: there are no peer-reviewed papers describing the use of Arezzo in a production environment, for example.) We point out that learning from successful (and unsuccessful) software development projects, and leveraging existing technology rather than attempting to reinvent it may help in making production-capable CGSs a reality.

The modified title of the paper now reflects this modified focus.

The body of our paper now begins by citing the Isern and Moreno review, lists the feature-set listed in section 3 of that review, (“Comparison”), and uses them as a framework to structure the rest of the paper’s Level 2 headings. Instead of re-advocating a requirement that has already been advocated (and sometimes implemented) by others, we take that requirement as a given (citing previous papers), and then discuss sub-requirements that follow from the broader requirement. For example, while all CGSs claim to support a repository, it is not clear whether features such as version control, secure access to authorized users, and privilege capability such as that supported by high-end DBMSs - e.g., read / write / execute, groups and roles, etc. are supported. (Only Vaidurya has documented search capability, and only HeCaSe2 has a security model.) All these sub-requirements become important in production systems.

**Specific Concerns:**

Abstract: **“The first section of the paper, says that most of the papers analyzed biases the requirements within the system/model proposed without a high-level perspective. This assumption is not correct.”**

We agree: the Isern and Moreno review provides the most direct counter-argument, as do papers 2-4 in the reviewer’s bibliography. We have removed these sentences in the abstract and the introduction.
The methods section should be improved giving more details about the databases used to collect papers, the coverage of the review, the criteria used during the search, papers considered, keywords used, topics discarded, etc.

We used the following databases (articles after and including the year 2000 only):

- PubMed, using the search term "practice guidelines as topic" [mesh] AND informatics AND "2000"[Publication Date] : "3000, which return 350 results (48 reviews), whose titles and abstracts were then used to filter to the topic of interest (e.g., omitting guideline development for specific diseases, since we focus on software).
- Inspec/Engineering Village (a computer science/engineering journals resource) was searched in two ways: a) "Clinical guideline" [title/subject/abstract] (for conference proceedings papers not in Pubmed) and b) “workflow”[title] and “requirements” [title/subject/abstract]. 1209 results were then filtered manually (by inspection of titles and then abstracts) for articles relevant to this paper’s theme.

Revisions: The above has been stated in the paper.

The authors propose the use of existing workflow-based suites to extend the use of CGs by other healthcare systems such as EMRs or CPOEs. This is an interesting idea for further systems but the authors do not discuss/analyze why current systems cannot be embedded into healthcare systems. For instance, some of these mentioned systems permit interaction with external elements by implementing interfaces described in a well-formed standard language as XML.

We were not very clear about this, and have now rewritten this text. We make the specific proposal that CGS implementers consider “compiling” the guideline to a standard workflow language like WS-BPEL (which is also an implementation of XML) in much the same way that Proforma compiles to Prolog, with calls to a custom library for guideline-specific tasks. The idea is that, as with the HeCaSe2 approach, the difficult problem of implementing an optimized, efficient execution engine is eliminated. Further, because of the large amount of vendor and third-party support for BPEL (notably in administrative systems, where it is heavily used by organizations like SAP, IBM, Oracle and HP) the task of interfacing would be simplified considerably. Similarly, verification and testing tools could be utilized without having to reinvent them. In other words, while it is possible to spend much effort implementing multiple interfaces, testing tools, etc. for a proprietary guideline language, it is simpler to leverage technologies with abundant vendor/third-party support.

While the designers of GLEE and other CGSs may have built interfaces, it is unclear as to how complete or error-free these interfaces are, since there are no peer reviewed publications describing their use with production EMRs in actual patient care. This would seem to indicate that the interfacing problem is much more difficult than commonly acknowledged. The problem is not solved simply by adopting XML: the difficulty lies in the comprehensiveness of the set of XML tags and attributes that a CGS’s implementers have defined to solve the interfacing problem. XML schemas can get extremely complex, based on the modeling problem to be addressed: for example, the schema for Microsoft’s Office Open XML (used in .docx, .xlsx etc. files of Office 2007 and later) contains over
3000 tags, and their meaning is almost impossible to understand without painstaking study of the ISO 29500 documentation (Microsoft’s own documentation is unusable).

**On pages 17-19, the authors discuss about the inference abilities required in a CG-based system.** The authors describe with an example that rule-based inference system are not able to interpret rules that include sonographic images (interpretable resources). Some recent works try to avoid this limitation including an intermediate layer between the CG and the clinician, such as an ontology. For instance, in [12] an ontology permits to map the results of a test (e.g. sonography) as interpretable attributes in a CG.

We were unclear, and have made the text more direct. The point we are making is that the CGS should be **inferencing-approach-neutral**, and allow the developer to use the approach that is most appropriate for a given sub-task (preferably reusing existing code – e.g., commercial or open-source libraries) instead of being forced to rewrite the algorithm using the guideline language.

We agree completely with the reviewer that ontologies greatly simplify the mapping problem. We now cite the above paper: we use a local ontology ourselves in mapping guideline elements to the corresponding data elements in the Henry Ford Health System EMR. However, inferencing – using a multivariate regression model or a support vector machine – and mapping are different, which require different approaches.

**In the same section, the authors explain that that those systems should be able to “explain” the actions taken.** Well, the authors can find some works about argumentation used in healthcare domain such as [13] and [14] that can solve their questions in this topic.

We agree. This was another instance of omitting the citation of previous work. We have now cited the above references, and shortened this sub-section dramatically.

**On pages 19-20, the authors distinguish three different modes of execution of CGs.** These three modes are quite subjective and can be fused in two modes: simulated and online. The first one can be used to test the performance and behavior of CGs using an interface and testing data, meanwhile the second mode uses real data and it is used on real time. The coverage of a CG (patients over a CG is enacted) can be one or more than one. This is the mode of operation on most of the existing guideline-based systems. Then, the proposed modes of operation should be explained and justified better with examples.

We have followed the reviewer’s recommendation and fused into simulated and online modes. This is a minor point, and so we have shortened the text. The reviews of de Clerq et al and Isern and Moreno point out the importance of verification and testing tools, and so we are not in disagreement.

**On page 20, performance efficiency and scalability are two different topics.** The authors discuss about the efficiency of execution of a CG but not on the scalability of a CG-based system.

We agree. We have defined scalability more accurately: the ability to deal concurrently with large numbers of patients, for multiple guidelines, at different stages of each guideline. (Therefore a process may be efficient, but not designed to scale.)
On page 21, the section implications of longer duration execution is irrelevant because the persistence of data is a mandatory requirement independently on the duration of the execution.

We were not clear on this. Using a DBMS for storage would address data persistence, but the problem is much more complex, requiring *an additional layer of software* that high-end commercial workflow engines (which typically utilize a DBMS-based persistence mechanism) implement.

We have now provided an actual example, in the revised text, reproduced below. The issues described below would not apply to a guideline-workflow whose temporal scope is brief (e.g., a single consultation).

“Unlike most decision-support systems, a patient going through a guideline workflow may stay within that workflow for long periods. For example, a workflow related to management of infertility would last more than a year, because pregnancy takes time to establish. The clinical aspects of the workflow may no longer apply (e.g., because of a successful new therapy, recommendations may have changed), and so a new clinical sub-workflow may supersede the old one. However, workflows also have administrative aspects (inventory adjustment, service itemization, reimbursement from the insurer/payer), whose management can be more complex: for different patients, multiple versions of the same administrative sub-workflow may run simultaneously. (For example, even though a reimbursement agreement has changed since the patient began treatment, the contract with the payer may require the old agreement to stay in force.)

“Business workflow engines address such situations through two means:

- Implementing a persistence mechanism that tracks workflow engine state, with details on every active instance – which workflow, which version, the current position within the version, the path taken to get to this position, etc.
- A sophisticated *version-aware* runtime engine.”

**References on CGs (theory and benefits) should be revised and improved drastically. For instance, recent works as [7], [15], and [16] permits to analyze the current use of CGs in daily care, identifying advantages, drawbacks and barriers that should be solved in the future.**

We have drastically shortened this section of the text. The target audiences of our manuscript presumably agree with both the reviewer and the authors that guidelines have important theoretical benefits. The above papers deal with how guidelines should be developed and evaluated, but this is, strictly speaking, a domain-expertise rather than an informatics problem. Our paper, as stated earlier, now focuses on the software aspects of implementation.