Step–By–Step Mark–Up of Medical Guideline Documents

Vojtěch Svátek and Marek Růžička

European Centre for Medical Informatics, Statistics and Epidemiology – Cardio
University of Economics, Prague, W.Churchill Sq. 4, 130 67 Praha 3
Czech Republic
svatek@vse.cz

Keywords: medical guidelines, XML, knowledge transformation, hypertension

Summary.

Approaches to formalisation of medical guidelines can be divided into model–centric and document–centric. While model–centric approaches dominate in the development of clinical decision support applications, document–centric, mark–up–based formalisation is suitable for application tasks requiring the 'literal' content of the document to be transferred into the formal model. Examples of such tasks are logical verification of the document or compliance analysis of health records.

The quality and efficiency of document–centric formalisation can be improved using a decomposition of the whole process into several explicit steps. We present a methodology and software tool supporting the step–by–step formalisation process. The knowledge elements can be marked up in the source text, refined to a tree structure with increasing level of detail, rearranged into an XML knowledge base, and, finally, exported into the operational representation. User–definable transformation rules enable to automate a large part of the process.

The approach is being tested in the domain of cardiology. For parts of the WHO/ISH Guidelines for Hypertension, the process has been carried out through all the stages, to the form of executable application, generated automatically from the XML knowledge base.

1. Introduction

Medical guidelines are standard means for dissemination of medical knowledge and for setting forth healthcare standards. Large attention is currently paid to their *formalisation* and subsequent *computational processing* both inside and outside the clinical environment. Different groups have developed their own repertories of formal guideline–modelling constructs and development methodologies.

Most guideline–computerisation projects are *model–centric*: a compact (often flowchart–based) conceptual model of the guideline is formulated by the domain expert in the early phase of the process, and gradually converted to a fully operational representation. The relationship between the original document and the model is only indirect, mediated by the expert, who is responsible for the initial 'text–to–model' leap. Since the conceptual model is semantically close to the operational model, it is relatively easy to proceed to a running application in this way. The model–centric approach has been repeatedly used for development of guideline–based decision–support systems, e.g. in *EON* [1], *Asgaard* [2], *GLIF* [3] or *Prodigy* [4].

An alternative stream in guideline computerisation is *document–centric*: the original text of the guideline is systematically *marked–up* with respect to the model and kept in the form of structured document (in the well–known XML – eXtensible Mark–up Language – format). The textual content thus evolves into some kind of guideline model more slowly, and the interpreting expert is more constrained by what the guidelines 'say literally'. The leader in this stream is the *GEM* methodology and model [5]; its authors claim that the mark–up–based approach is more appropriate for capturing (in addition to decision structures) the 'healthcare–service' aspects of the guideline, such as its prospective audience or support with clinical evidence.

Mark–up–based formalisation naturally leads the developers to structure the documents down gradually, in multiple steps. In contrast to *ad hoc* phasing recognised as important in GEM [5], we advocate a more systematic approach. We assume that a methodology designed for a given formalisation problem, with *explicitly* defined steps, makes the process more transparent. Thanks to the limited number of transformations performed in each step, the risk of information loss is reduced and subsequent verification is made easier. These features are particularly important when we prefer to preserve the generic (and, as much as possible, 'literal') content of the guidelines rather

than to adapt them to local conditions. This situation is typical for certain guideline–related tasks:

- *Formal verification* of textual guidelines [6] based e.g. on temporal logics.
- Empirical *compliance analysis* [7]: large–scale comparison of the actual medical practice (reflected in electronic patient records) with standards set by the guidelines.

A common feature of both tasks is their execution outside the clinical environment, which relaxes the safety criteria (imposed by online decision support), alleviates the ultimate responsibility of the physician, and thus gives way to fully autonomous (non–interactive) computational means.

2. Methodology of the Step–By–Step Approach

The transition from a plain text document containing knowledge to an operational representation includes multiple aspects:

- *Generic linguistic expressions* expressing e.g. the structure of definitions, decisions or causalities (such as 'if...', 'but...', 'provided...', 'should be...', 'is always...') have to be replaced with *standardised formal structures*
- Free text that cannot be formalised or is irrelevant to the task has to be *removed*
- Knowledge elements have to be *modularised*, i.e. made independent of the surrounding context in the document
- Free–text terms referring to the same *domain concept* have to be replaced with standard *vocabulary terms*
- *Missing* (implicit, background...) knowledge has to be added
- *Vague* formulations reflecting the state of measurable parameters (such as 'good response to therapy' or 'several times a day') have to be replaced with concrete values.

There are several plausible ways how to map these different aspects onto a sequence of steps: the one we propose here assumes six levels of formalisation. Each level (except for the last) is, in practice, a specialised XML–based language, and has its own DTD (Document Type Definition).

1. *Input text format*. The natural choice is *XHTML* [9], the XML version of HTML (HyperText Mark–up Language). The creation of an XHTML document merely

requires common web page design skills; the documents can be viewed with web browsers, and their elements can be referenced using the XLink [10] technology.

2. *Coarse–grained semantic mark–up*. Large (from sentence–level up) and relatively closed chunks of text (or tables) are semantically marked–up, and parts of the document that do not have operational semantics are removed. We assume that coarse–grained mark–up can be done by persons without deep medical expertise.
3. *Fine–grained semantic mark–up*. The basic elements are refined into a tree structure of sub–elements. The stream of free text is thus disconnected, its parts however remain in the original order wherever possible. Linguistic reformulation is often needed in order to pick up relevant phrases consistently out of a complex sentence. Background knowledge is added (by expert–physician) so as to resolve ambiguous statements and to provide the missing aspects of knowledge elements. A *data dictionary* is created, which characterises the important clinical parameters involved e.g. in decision structures and concept definitions.
4. *Universal knowledge base*. The original document structure is abandoned in favour of systematic ordering. The context of occurrence of the knowledge elements is 'wrapped' into their own structure, to achieve modularity. Cross–references in the text are verified and updated if necessary. Some of these activities can be done by the knowledge engineer (software support being almost inevitable!), while other require the involvement of medical expert.
5. *Export–specific knowledge base*. In this last level of XML mark–up, the structure of elements is adapted so as to ease the export to the target representation (even at the expense of readability). The same universal knowledge base thus can be used to create different target representations thanks to this 'intermediate' level.
6. *Target computational representation*. The ultimate format can be either that of an operational knowledge–engineering environment (in our project we consider OCML [11] and Prolog) or even a conventional programming language (here, the class structure in Java). The export can be carried out fully automatically, using the declarative apparatus of XSL [12] style sheets.

We tested the approach using a simple guideline model with four top–level elements: *procedural* statements—which are, at latter stages, refined to *scenarios*—definitions of and references to *concepts*, *goals* to be achieved, and *causal*

relationships. The heart of computational processing are procedural scenarios systematically conditioned by expected history of treatment, as a sort of compromise between compact ('flowchart') procedural modelling of e.g. GLIF [3] and stand–alone Medical Logic Modules (MLMs) [13]. The model has, in each of the levels of formalisation, a different shape: the elements evolve from free–text containers through thoroughly marked–up text into an XML knowledge base and, finally, into the computational representation (see [14] for details). However, though the model appeared viable in our experiments, we intend to adopt a more complex model (allowing e.g. guideline branching and nesting) as soon as the step–by–step technology itself is fine–tuned.

3. Tool Support

The beta version of the dedicated step–by–step mark–up processor has been developed (in Java) under the name of *Stepper*, with the following main functionalities:

- Support for the mark–up of knowledge elements *in source text*, incl. specification of their *attribute values*.
- Fully automated generation and update of element–to–text and element–to–element *links* across the formalisation levels, and *retrieval* of knowledge elements arisen from the given text fragment and vice versa.
- Convenient creation and update of *transformation rules*, which enable to define operations such as element aggregation, decomposition, shift of element content into attribute, and even conditional setting of element value.
- Offline export to non–XML format via an integrated XSLT [12] processor.

As soon as the transformation rules have been defined, the users can carry out the initial mark–up and move information around the XML (tree) structures automatically built by the rules. When proceeding from one level of semantic mark-up to another, the screen is horizontally divided into two parts, source and target. Each consists of an XML tree and a pane for editing attribute values; tree structures, buttons and attribute–value forms are generated in runtime from the DTD of the given formalisation level.

4. Application

We tested the methodology on the *WHO hypertension guidelines* [15], in the context of the European project 'Medical Guideline Technology', in 2000–2001. The document mark–up was basis for the development of a compliance–analysis (and partially also decision–support) application [16]; the target language was OCML (Operational Conceptual Modelling Language [11]). Since the formalisation had to be carried out manually (the first version of the *Stepper* tool appeared as late as in Autumn 2001), only the first three levels have been achieved completely, and the target application was thus based only indirectly on the semantic mark–up of the document.

Currently, in the EuroMISE Centre – Cardio (a new national–level research centre), we are both revisiting the hypertension application (with the help of the *Stepper* tool), and starting to address another cardiolological application, namely, unstable angina. For hypertension, formalisation of selected parts of the guideline has been led through all the steps mentioned in section 2. The result is a collection of simple interactive applications *automatically generated* (in Java) from the XML knowledge base.

Fig. 1 displays a series of snapshots of the formalisation process. The XHTML source text is displayed in the *Stepper* window (upper–right) in the same way as in a web browser. The user marks up a piece of text and denotes it as *procedural*. In the 'Text' card of the lower–right pane (not shown), s/he further elaborates the 'mixed content': `<con>` elements (denoting potential concept references) are marked up within the text. For the next step, transformation rules prescribe that the content of *procedural* will be transferred to *scenario*/s and that the concepts should be captured in the *data dictionary*; 'drug tolerability' will be declared as *ordinal*. When proceeding to 'Level 2', the precondition of scenario ('treatment has already started') is formalised further: the implicit notion of 'monotherapy' is introduced, and free text transformed to simple formal statements (predicate–value pairs). Finally (having skipped the level of 'export–specific knowledge base'), we proceed via XSLT to the operational Java code.

Fig. 1: Example of formalisation process

5. Related Work

Our *step–by–step* methodology is not the only one within guideline formalisation research. For example, [17] declares the stepwise character of formalisation in GLIF and in MLMs. The main difference from our approach is the *top–down* character of formalisation: the model already formulated by the expert (in flowchart form in GLIF and in textual form in MLMs) serves as starting point, and is merely refined in the subsequent steps. In contrast, our document–centric approach is *bottom–up*: the most important part of the formalisation happens before some sort of self–contained model (namely, the universal knowledge base) comes into being.

Our stress on keeping track of 'text–to–model' transformations has its counterparts in the use of *mark–up technology* for model–centric formalisation. Examples of recently developed 'model–centric' mark–up tools are *GMT* [18] and *DeGeL* [19]. Both have been developed in connection with the *Asbru* guideline model and language, and offer some level of model–to–text linkage. Particularly interesting is then the work by Shankar [20], who criticises the *rigidity* of straightforward linkage, and attempts to overcome it by Information Retrieval methods. Instead of absolute addresses, the model elements are associated with conceptual descriptions that enable to retrieve the relevant portions of the document dynamically. Note that our solution to essentially the same problem consists in 'segmenting' each text–to–model link into multiple parts (corresponding to transformation steps): if the document (or even the model) changes, only the adjacent part of the link has to be modified. We thus remain faithful to the document–centric paradigm while eliminating one of its drawbacks; conversely, Shankar's approach stands on the model–centric ground.

6. Discussion

Among the six aspects of text formalisation mentioned in section 2, the technology of stepwise mark–up directly handles the first three: transformation of guideline elements (decisions, definitions...) into the semi–formal structure, removal of useless text, and modularisation of elements (via enrichment by 'context' information). The remaining three—shift to controlled vocabularies, addition of implicit knowledge and handling vague formulation—require further effort to be tackled satisfactorily.

Adoption of controlled vocabularies such as *ICD–10* or *SNOMED* would be rather straightforward. Although we have not implemented vocabulary linking in our tool yet, it could be part of fine–grained semantic tagging. In our guideline model, the element for ‘concept definition‘ already contains sub–elements for ‘canonical‘ name and for ‘aliases‘, which can be used for distinguishing vocabulary terms from *ad–hoc* ones.

Addition of external knowledge is a thorny issue, since we could easily end up with a model semantically different from what was *intended* by the guideline authors. It is to be determined whether a piece of knowledge has been omitted (or left in vague form)

- by mistake
- as part of ‘basic‘ medical knowledge every physician should be aware of
- due to lack of consensus in the guideline–authoring body itself
- due to dependency on the local conditions of treatment (e.g. availability of drugs).

Our method currently only offers *passive* ‘track–keeping’ approach. We used an XML attribute ‘added’ (shared by all top–level elements) with allowed values: *no* – text without modification, *interp* – text has been reformulated using the most likely linguistic interpretation, *parts* – part of the text has been added, and *whole* – the whole element has been added. This enables to identify the potential entry points of subjective information but does not solve the formalisation problem itself. *Active* methods should, for example, try to eliminate the subjectivity of added knowledge via enquiries to *multiple independent experts* (ideally, physicians associated with the authoring body). Subsequently, *fuzzy measures* could be a technical means for aggregating different opinions of experts (on quantitative parameters) into an operational representation.

Another point that could possibly be argued is the ‘selective demand for expertise’ along the formalisation steps. Patel [21] has proven in the model–centric setting (GLIF) that joined effort of domain experts and knowledge engineers leads to improved results. In our methodology, two of the transformation steps (fine–grained mark–up and construction of universal knowledge base) rely on such synergy. Since the ultimate steps are more–or–less mechanical, it is only the very first step (coarse–grained mark–up) which is attributed to the knowledge engineer alone. This is based on the facts that classification of high–level knowledge blocks has little to do with ‘real’ medical expertise (it is rather based on common–sense reasoning over generic linguistic

constructs), and, if a mismatch still occurs, it would probably be identified in the next step when the same blocks are to be refined.

One of advantages of document–centric approaches is the possibility to maintain *different parts of the document* in different levels of formalisation, as mentioned in *GEM* [5]. This, in a sense, goes well together with our step–by–step view, since explicit formalisation levels can be more easily separated than *ad–hoc* (implicit) ones. On the other hand, the *Stepper* tool does not (by its nature) support the mixing of different levels in the same document. Rather, the documents in later stages of formalisation may contain only some parts of the documents in earlier stages of formalisation.

7. Conclusions

In the paper, we have described the methodology, model and software tool for step–by–step transformation of medical guideline documents into a formal (or even operational) representation. The explicitly defined levels guide the whole process and help to minimise information loss. The approach seems to be particularly adapted to the situation when the guideline text is mostly *narrative* (rather than flowchart– or table–based) and we want to keep its *literal* content, be it for stand–alone verification or for compliance analysis over patient records.

Future work will, among other, address the capture of *consensual* background knowledge needed to operationalise vague statements and to fill gaps in knowledge due to implicit knowledge assumptions. Attention will also be paid to overcoming some technical limitations of the first version of the *Stepper* tool, and to the evaluation of other existing guideline models in the step–by–step framework.

Acknowledgements

The authors wish to express their thanks to Tomáš Kroupa, who contributed to the development of the mark–up languages, to the medical expert Jan Peleška, to Jana Zvárová, Director of the EuroMISE Centre, and to an anonymous reviewer, for inspiring comments on earlier drafts of this paper. The research has been partially supported by the project *LN00B107* (European Centre for Medical Informatics, Statistics and Epidemiology — Cardio) of the Ministry of Education of the Czech Republic.

References

[1] M.A. Musen, S.W. Tu, A.K. Das, and Y. Shahar, EON: A Component–Based Approach to Automation of Protocol–Directed Therapy, JAMIA 3:367–388, 1996.

[2] Y. Shahar, S. Miksch, and P. Johnson, The Asgaard Project: A Task-Specific Framework for the Application and Critiquing of Time-Oriented Clinical Guidelines. Artificial Intelligence in Medicine 14:29-51, 1998.

[3] M. Peleg, A. A. Boxwala, O. Ogunyemi, Q. Zeng, S. W. Tu, E. Bernstam, L. Ohno–Machado, E. H. Shortliffe, and R. A. Greenes: GLIF3: The Evolution of a Guideline Representation Format. AMIA 2000 Annual Symposium, Los Angeles, CA, 645–649.

[4] P. D. Johnson, S. Tu, N. Booth, B. Sugden and I. N. Purves: Using Scenarios in Chronic Disease Management Guidelines for Primary Care. AMIA Annual Symposium, Los Angeles, CA, 389–393. 2000.

[5] R. N. Shiffman, B. T. Karras, A. Agrawal, R. Chen, L. Marenco and S. Nath: GEM: A proposal for a more comprehensive guideline document model using XML. JAMIA 2000; 7(5):488–498.

[6] M. Marcos, M. Balser, A. ten Teije, and F. van Harmelen, From informal knowledge to formal logic: a realistic case study in medical protocols, in: (A. Gomez–Perez and R. Benjamins, eds.) Proc. 13th Int. Conf. on Knowledge Engineering and Knowledge Management, LNCS, Springer–Verlag, 2002, 49–64.

[7] A. Říha, V. Svátek, P. Němec, and J. Zvárová, Medical guideline as prior knowledge in electronic healthcare record mining, in: Data Mining III. (Eds. Zanasi A., Brebbia C.A., Ebecken N.F.F.E.,Melli P.), WIT Press, Southampton, 2002, 809–818.

[8] B. Seroussi, J. Bouaud, E.–C. Antoine, L. Zelek, M. Spielmann, An experiment in sharing and reusing OncoDoc's breast cancer guideline knowledge. Computer–Based Support for Clinical Guidelines and Protocols, Studies in Health Technology and Informatics, Vol.83, IOS Press, 2001.

[9] M. Altheim, S. McCarron, XHTML 1.1 – Module-based XHTML – W3C Working Draft. W3C 2000, `http://www.w3.org/TR/xhtml11`.

[10] S. DeRose, E. Maler, D. Orchard, and B. Trafford, XML Linking Language

(XLink), W3C Working Draft, W3C, 1999, http://www.w3.org/TR/xlink.

[11] E. Motta, Reusable Components for Knowledge Modelling: Principles and Case Studies in Parametric Design, IOS Press, Amsterdam, 1999.

[12] J. Clark, XSL Transformations (XSLT) Version 1.0, W3C, 1999. http://www.w3.org/TR/xslt.

[13] G. Hripcsak, P. Ludemann, T. A. Pryor, O. B. Wigertz, P. D. Clayton, Rationale for the Arden Syntax. Computers and Biomedical Research 1994;27:291–324.

[14] V. Svátek, T. Kroupa, and M. Růžička, Guide–X – a Step–by–step, Markup–Based Approach to Guideline Formalisation, in (B. Heller , M. Loeffler, M. Musen, M. Stefanelli, eds.:) Computer–Based Support for Clinical Guidelines and Protocols, IOS Press, Amsterdam, 2001, 97–114.

[15] WHO/ISH Guidelines for the Management of Hypertension. Journal of Hypertension, 17, 1999, 151—183.

[16] V. Svátek, A. Říha, T. Zíka, J. Zvárová, R. Jiroušek and Z. Zdrahal, Informal, Formal and Operational Modelling of Medical Guidelines, in (Hruška T., Hashimoto M., eds.:) Knowledge–Based Software Engineering. IOS Press, 2000.

[17] M. Peleg, A. A. Boxwala, E. Bernstam, S. Tu, R. A. Greenes, E. H. Shortliffe. Sharable Representation of Clinical Guidelines in GLIF: Relationship to the Arden Syntax. J. Biomedical Informatics Vol. 34, No. 3, June 2001, pp. 170-181. 2001.

[18] R.D. Shankar, S.W. Tu, S.B. Martins, L.M. Fagan, M.K. Goldstein, and M. A. Musen, Integration of Textual Guideline Documents with Formal Guideline Knowledge Bases, in Proc. AMIA 2001.

[19] R. Kosara, S. Miksch, A. Seyfang, and P. Votruba, Tools for Acquiring Clinical Guidelines in Asbru, in Proceedings of the Sixth World Conference on Integrate Design and Process Technology (IDPT'02).

[20] Y. Shahar, A Hybrid Framework for Representation and Use of Clinical Guidelines, in Proc. AMIA 2002, San Antonio, Texas 2002.

[21] V. L. Patel, V. G. Allen, J. F. Arocha, & E. H. Shortliffe. Representing Clinical Guidelines in GLIF: Individual and Collaborative Expertise. Journal of the American Medical Informatics Association 5(5):467-83, 1998.