Easing the Formalization of Clinical Guidelines with a User-tailored, Extensible Agile Model Driven Development (AMDD)

Patrick Martini, Katharina Kaiser

Vienna University of Technology

Inst. of Software Technology & Interactive Systems

Favoritenstrasse 9-11/188, 1040 Vienna, Austria

{martini, kaiser}@ifs.tuwien.ac.at

Silvia Miksch

Danube University Krems

Dept. of Information and Knowledge Engineering

Dr.-Karl-Dorrek-Strae 30, 3500 Krems, Austria

silvia.miksch@donau-uni.ac.at

Abstract

Transforming a text-based clinical guideline in a computer-interpretable form is a time-consuming and demanding task due to the various users involved, who have different technical and medical background. In the past, different guideline representation languages and supporting tools have been developed, however, these approaches seldom address the various users' demands and needs in the different steps of the guidelines' life cycle. Our approach is oriented on the guideline life-cycle and takes the requirements and the interactions of the various actors into account to formalize guidelines in a computer-interpretable guideline representation by using a semi-automatic way based on NLP techniques. We analyzed the guideline life-cycle and the roles of the actors to build such a model. This model is prototypical implemented and showed the usefulness and utility for the various users.

1. Introduction

The evidence-based medicine/healthcare (EBM) [11] movement has been gaining ground quickly over the past few years promoted by politicians, clinicians, and management in order to improve the quality and consistency in healthcare reducing costs where possible. EBM also influenced the development of clinical practice guidelines (CPGs) – also known as medical guidelines – to support quality assurance of CPGs compared to the consensus-based development.

The well-known definition of CPGs states that they are "systematically developed statements to assist practitioner and patient decisions about appropriate health care for specific clinical circumstances" [4]. Such guidelines are designed to assist health care practitioners in the prevention, diagnosis, treatment, and management of specific clinical conditions.

CPGs, used correctly, have been proven to be capable of improving patient care [7] and supporting health care practitioners in their decision-making process. However, CPGs are often characterized by their limited use in practice: rarely, practice behavior has been changed by them [19]. As analyses have shown, passive dissemination of guidelines (e.g., publication in a medical journal) has not been successful in changing behavior of the medical staff. However, if additional active implementation strategies have been used, then one can observe changes in practice patterns [3].

Hence, the new focus of EBM is directed towards the development of effective strategies for translating evidence into practice.

This is a process that involves different academic disciplines, computer-science included. Therefore, we have to support this process by developing a system that backs the healthcare practitioners on implementing and on using the CPGs in routine clinical practice.

Our aim in this process is to formalize the guidelines in a computer-interpretable guideline representations. Even though, several groups have hitherto focused on this subject (an exhaustive overview of different approaches can be found in [13]), the majority of the related approaches were developed before introducing EBM. Consequently, firstly, they cannot cover all the facets peculiar to EBM-oriented guidelines (e.g., strength of recommendation, level of evidence), secondly, they cover only the executing part forgetting the other elements that contribute to the medical epistemology, and third, they accomplish their task with a manual modeling which is error prone and hugely time-consuming.

Hence, our approach takes these characteristics better into account using a semi-automatic way based on a Natural Language Processing (NLP) techniques. In order to address the problem, we build different components that operate in a pipeline fashion (see Figure 1). This pipeline takes a guide-
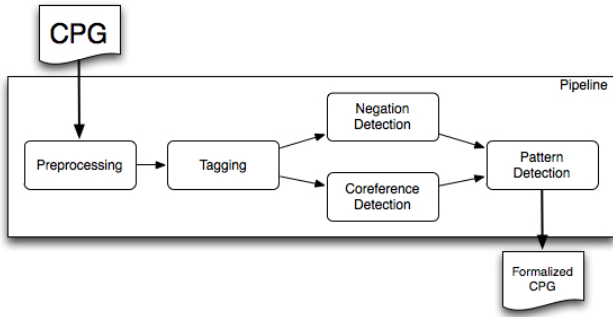
Figure 1. Architecture of the NLP pipeline for medical guidelines.

line as input and produces as output a formalized version in which the execution part could be easily translated into other guideline representation languges, such as PROForma or Asbru. A very important part is that the correctness of each step of the pipeline is evaluated before the formalization process can go further. An error in one step can easily propagate through the pipeline with dramatic effects.

Regarding the formalization process, initially we focus on the identification of the actors, the phases, and the interactions involved in the life-cycle of a guideline. In the previous scenario the interaction becomes crucial.

Consequently, we need to evaluate how we can support the formalization process, and at the same time, can render the interaction between the different actors as smooth as possible. Shedding light on these issue is the main contribution of this paper.

In the next section, we give an overview of the current related approaches. In Section 3, we present our contribution. A prototype implementation of our approach can be found in Section 4 followed by conclusions.

2. Related Work

Currently, there are two kinds of approaches to model computer-interpretable CPGs: (1) model-centric techniques and (2) document-centric techniques. In the former, a conceptual model is formulated by domain-experts. Thus, the relationship between the model and the original document is only indirect. Representatives of this techniques are *AsbruView* [9], *Protégé* [6], *Arezzo*, and *Tallis* [17]. In document-centric approaches markup-based tools are used to systematically mark-up the original guideline in order to generate a semi-formal model of the marked text part. Representative tools are the *GEM Cutter* [14], *Stepper* [15], the *Document Exploration and Linking Tool / Addons (DELT/A)* [18], *Uruz*, part of the *Digital electronic Guideline Library (Degel)* framework [16], *Clinical Prac-*

tice Guideline Reference Architecture (CPG-RA) the revisited version of the previous CPGA [2].

Most of these approaches accomplish their task in a manual manner and they concentrate on the executable part as we stated before. We try to disburden the involved users by automating parts of the formalization process and taking in high consideration the physicians' perspective. In the next sections we present our approach in more detail.

3. Our Approach

3.1. Main Idea

Due to the sensitive nature of a guideline, the formalization process requires an intensive and continuous one-to-one collaboration between the knowledge engineers and the healthcare practitioners. Therefore, we need a system that can easily integrate the new insights coming from anyone involved in the process and permits to evaluate each step of the pipeline easily. For the development of such a system, we take inspiration from the concepts behind the Agile Model Driven Development (AMDD) [1, 5].

At the bottom of the AMDD the concept of the model is located (more precisely, we speak about meta-model, but this refinement goes beyond the scope of this paper): a model is a representation in which each element of the representation corresponds to an element or a concept in the domain. For us, models are attractive as they use terms and concepts familiar to people who work in the domain to describe the problem - unlike those terms which hitherto have only been familiar to knowledge engineers.

Hence, we start to think of a model that can represent a guideline analyzing its domain and its aspects. The two fundamental aspects we distinguish are: (1) the life-cycle of a guideline and (2) tool friendliness [10].
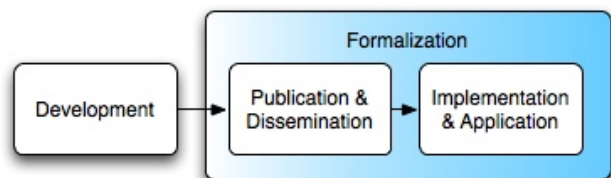

Figure 2. The life cycle of a guideline.

Guideline Life-Cycle

Studying the life-cycle of a guideline permits us to identify (1) which are the phases, (2) who are the actors, and (3) what are the consequent interactions and the overlaps between them.

We consider the following phases important in the study of a life-cycle (see Figure 2):

1. **Development**. At this stage the medical guideline is created.

2. **Publication and Dissemination.** After the *development phase* the medical guidelines have to be published starting their problematic arduous dissemination path.

3. **Formalization.** It is the phase in which the knowledge engineers turn the narrative text of the guideline into a computer-interpretable version.

4. **Implementation and Application.** The guidelines are introduced into the healthcare environment and implemented in the active use of the formalized version during the daily healthcare activity.

We have found three main actors which are actively involved in the life-cycle of a guideline:

1. **Guideline developers**. They are grouped in multidisciplinary, national representative groups, who are responsible for the content of medical guidelines (development phase), as well as their publication and dissemination.

2. **Healthcare practitioners**. Many healthcare professionals with different profiles fall within this category. Due to the varying complexities of guidelines, we decided to consider mainly physicians even though we do not underestimate the other groups.

3. **Knowledge engineers.** They are computer science experts who have the duty to formalize the guideline.

Regarding interactions and overlaps, Figure 2 shows that the formalization phase overlaps with the implementation and application phase but we also think that an overlapping with the publication and dissemination phase is possible. This overlapping is crucial: knowledge engineers and physicians do have to interact with each other. However, this synergy is only possible when there is no element hindering these kinds of communication: e.g., as each actor has her/his own language, the model should function as a kind of translator.

Tool Friendliness

Another important aspect to take in consideration when modeling a domain is the "tool friendliness". The term "tool friendliness" refers to how easy it is to design and to develop a tool for a specific model. Being "tool friendly" is not a scientific measure: it is a way to express the quality of a model from a tool development aspect [10]. We focus principally on the end user of a complete formalized guideline - healthcare practitioners - in order to find out which are the functionalities helping them to benefit from the use of guidelines. As a logical consequence this could help the dissemination phase in the end.

Interviewing physicians, who participated at the GIN Conference in Budapest 2006, it emerges that a particular focus needs to be put on the application phase in order to obtain this particular outcome for the dissemination. So far, this former phase is characterized by three different stages:

1. **Screening**. The screening phase consists of those kinds of operations that permit us to understand if the guideline is exactly what we are looking for. This process includes activities, such as summarization and so on.

2. **Knowledge Acquisition**. The knowledge acquisition phase entails gleaning clinical knowledge from the narrative text of a guideline in order to consolidate the personal knowledge on the particular guideline topic.

3. **Employment**. The employment phase starts after a consistent knowledge acquisition in which the physicians use the guideline actively during their daily activity.

However, one major problem persists: when it comes to reading, one can observe that computer users prefer to read texts from paper than off a computer screen. Hence, we need to consider in that computer applications must be changed in such a way that they support knowledge acquisition through reading. By doing so, their role of "just displaying" can be extended. Therefore, we have to support the previous stages with the following features:

- **Non-linear navigation**. It permits users to scan through the guideline text jumping easily from a point of text to another, supported also by text visualization.

- **Annotation**. Analyzing a pen-paper based reading [12] we realize that we have to introduce the typical functions of an old fashion style reading, such as free marking, annotation, and notes. These kind of elements can stay at a text level or they can augment the guideline model becoming an integral part of it.

- **Visualization methods**. They allow to display the model, which can then easily be understood by the physicians and give them the possibility to access and interact with the guideline in another way. Such visualization methods could be, for instance, tree-like, mindmap-like.

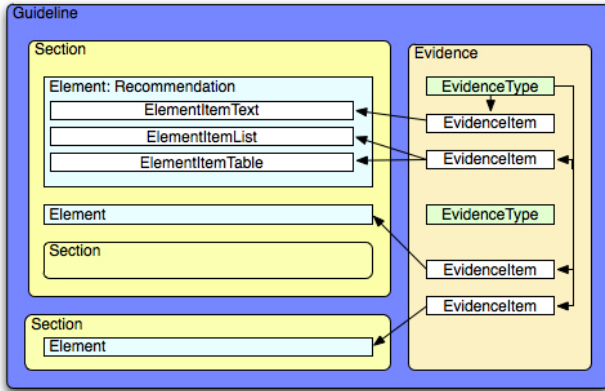All these features have to complement each other as smooth as possible.

Figure 3. A simplified schema of the structure of a guideline in the publication phase.

Another important aspect is the physicians' different medical background: what is of vital importance to someone is trivial to another one. Ignoring a priori the physicians' background, the model has to capture all information available in a guideline. This can be boiled down to a granularity problem. The information included in a guideline has one or more levels of granularity providing a function that permits the user to set the proper level of information in accordance with their dynamically changing necessities (e.g., it is obvious that after a consistent knowledge acquisition, in the employment phase the quantity of information necessary to follow up the guideline is less).

3.2. Guideline Model

Bearing in mind the considerations of the previous section, we now describe the guideline model that represents a guideline after the publication phase. The model has to satisfy the following requirements:

- **Extensibility**. We imagine our model like a sponge, capable of absorbing easily new elements coming from the different formalization process steps.

- **Flexibility**. The model should be easily adaptable to a changing guideline domain: our model should evolve hand in hand with the life-cycle of the guideline.

- **Tool friendliness**. The model should account for the tool requirements surfaced in the previous section, thus assigning an important role to the end users.

Normally, when speaking about modeling two fundamental approaches come up: the bottom-up approach and the top-down approach. Both approaches have advantages and disadvantages (for a general overview see [10]). However,

combining them we obtain an approach called "meet in the middle" [10] that provides us with the best of the two "worlds". In this approach, development starts from two directions (bottom and top): firstly, taking in consideration the functionality of a hypothetical tool and secondly, at the same time analyzing the guideline in order to extract the concepts and the semantics that are going to form the guideline domain. With the functionality of a hypothetical tool we have dealt in the previous section, hence, we will now focus on the analysis of the guideline domain.

The form in which the information is presented is important as it conceals an implicit semantic (e.g., list, paragraph) that we have to preserve in our model. We are going to explain our model by starting from the atomic element up to the composite ones (compare Figure 3):

ElementItemText. It represents the smallest element in which we can divide a guideline (atomic element). At the lowest level, we observe a guideline to be formed by a chunk of text somehow combined. At this stage, we have matched it to paragraph level. In future, this level could be refined to sentence level, up to word level, if the formalization process requires it.

ElementItemList, ElementItemTable. As we can deduce from the name they are used to model lists and tables, respectively. They can contain both *ElementItemText* and *Element*. The latter is used to model a nested structure, such as nested list and so on.

Element. Sometimes, small parts of a guideline need to be grouped together. For example, a recommendation can be composed by different elements (text, list, etc.). We have to keep all these elements together in order not to lose information. For instance, a recommendation is composed by different elements that must stay together. The Element entity can contain *ElementItemText, -List, -Table*, and *Element* in turn (in a recursive way).

Section. Each guideline is composed by one *Section* at least. The sections are used to divide the different topics presented in a guideline. A *Section* can contain a *Section* in turn (in a recursive way) and *Element*.

Guideline. This is the main container of the previous elements together with the evidence elements that we are going to explain later.

Other important elements of a guideline are the evidence information and its schema. As every organization has its own system [8], we decided to additionally represent the schema and the definition of the evidence in order to be "evidence system grade independent". To accomplish this aim, we use three entities:
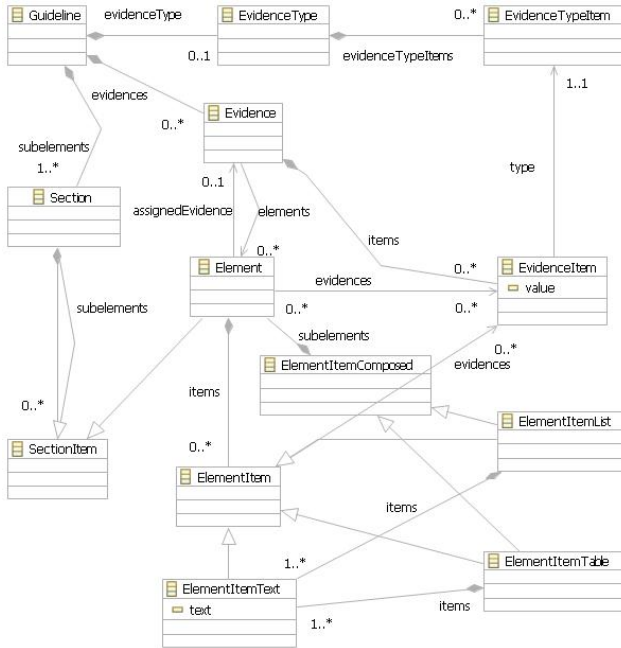
Figure 4. The model expressed with Ecore.

EvidenceType models all element types composing the evidence presented in a guideline, such as level of evidence, strength of recommendation, benefit, and so on.

EvidenceItem models all values of the previous elements. It allows to associate the level of evidence with its value (e.g., level of evidence 1++).

Evidence is a container of *EvidenceItem*. It can also be used to give more emphasis on a particular combination of *EvidenceItems* (e.g., level of evidence 1++; strength of recommendation A).

In general, the evidence information can be associated with the following elements: *Element*, *ElementItemText*, *ElementItemList* and *ElementItemTable*.

Furthermore, the model can be expanded in two directions: *horizontally* and *vertically*. Vertical expansion is taking place when new elements can give to the guideline a new formalization form (the section is divided into actions, etc.). Then, we can state that the granularity is growing. Horizontal expansion is given when new elements insert new information not present in the original guideline (e.g., physicians' annotation).

4. Prototype Implementation

For our prototype implementation, we selected 25 guidelines for the treatment of cancer from the National Guideline Clearinghouse (NGC), a public resource for evidence-based clinical practice guidelines. These guidelines were
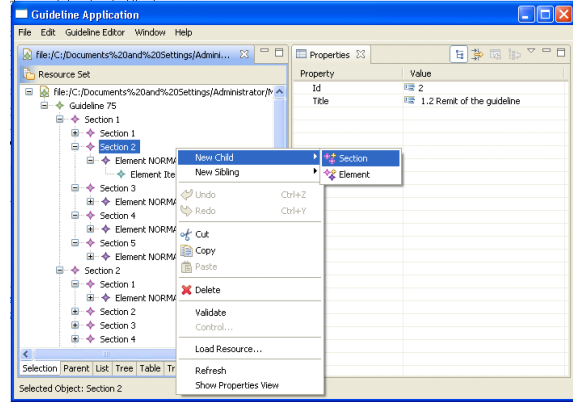

Figure 5. The EMF editor in action on a formalized version of Epithelial ovarian cancer guideline.

developed by different organizations with different formats and different evidence schemas, in order to prove that all these guidelines are representable with our model.

We built a prototype to test our model by using Eclipse Modeling Framework (EMF) and Graphical Modeling Framework (GMF) technologies of the Eclipse Workbench. EMF is a Java open-source framework and code-generation facility for building tools and other applications based on a structured model. While the Eclipse Platform provides a powerful integration framework at the UI and file level, EMF enhances this capability to enable fine-grained data sharing among tools and applications. The core EMF framework includes a meta model (Ecore) for describing models and runtime support for the models. GMF provides a framework for developing tools and diagram editors based an EMF model. Natively, they provide two different views – tree-like and mindmap-like – together with the annotation feature that is going to extend the model.

Firstly, we defined our model in the Ecore syntax (see Figure 4). Secondly, we filled our model with the information extracted by a Ruby-based web-scrapper that exploit the HTML guideline format to identify all elements composing a guideline. The model can be filled manually as well, by using the EMF Model editor (see Figure 5) that enables the employment of our model already in the *Publication and Dissemination phase*.

5. Conclusion

Eliciting the semantic and conceptual content of text in explicit representation is a way forward to attribute to the computer a more active role. We consider our model a good starting point for a guideline formalization process. It can be extended easily and hence, support the entire formaliza-

tion process.

The adoption of the modeling approach "meet in the middle" allows us to gather a lot of information reusable for future work, like the implementation of a tool that supports the life-cycle of a guideline and its actors.

Although our implementation is only a prototype, it has proven to us that we are on the right track. Our model can be used by different actors: the developers can use it for publishing, the physicians for medical epistemology and employment, and the knowledge engineers for supporting the NLP pipeline evaluating each step in interaction with the physicians.

In future, the model will be extended following our pipeline. In the final stage, the model can be easily translated also into other languages, such as Asbru and PRO-Forma.

References

[1] S. W. Ambler. Agile model driven development (AMDD): The key to scaling agile software development, November 2007. http://www.agilemodeling.com/essays/amdd.htm.

[2] S. Chu. Guideline representation formalism and electronic decision support systems: Addressing the guideline – implementation gap. *Health Care and Informatics Review Online*, March 2005.

[3] D. A. Davis, M. A. Thomson, A. D. Oxman, and R. B. Haynes. Evidence for the effectiveness of CME. A review of 50 randomized controlled trials. *JAMA*, 268(9):1111–1117, 1992.

[4] M. J. Field and K. N. Lohr, editors. *Clinical Practice Guidelines: Directions for a New Program.* National Academies Press, Institute of Medicine, Washington DC, 1990.

[5] M. Fowler. The new methodology, December 2005. http://martinfowler.com/articles/newMethodology.html.

[6] J. H. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubézy, H. Eriksson, N. F. Noy, and S. W. Tu. The Evolution of Protégé: An Environment for Knowledge-based Systems Development. *International Journal of Human Computer Studies*, 58(1):89–123, 2003.

[7] M. E. Johnston, K. B. Langton, R. B. Haynes, and A. Mathieu. Effects of computer-based clinical decision support systems on clinician performance and patient outcome: a critical appraisal of research. *Ann Intern Med*, 120(2):135–142, 1994.

[8] K. Kaiser, P. Martini, S. Miksch, and A. Öztürk. A meta schema for evidence information in clinical practice guidelines as a basis for decision-making. In K. Kuhn, T. Y. Leong, and J. Warren, editors, *Proceedings of the 12th World Congress on Health (Medical) Informatics (Medinfo'2007)*, pages 925–929, Brisbane, Australia, 2007. AMIA.

[9] R. Kosara and S. Miksch. Metaphors of Movement: A Visualization and User Interface for Time-Oriented, Skeletal Plans. *Artificial Intelligence in Medicine, Special Issue: Information Visualization in Medicine*, 22(2):111–131, May 2001.

[10] P. Kovari. Explore model-driven development (mdd) and related approaches: Applying domain-specific modeling to model-driven architecture, Sep 2007. http://www.ibm.com/developerworks/library/ar-mdd4/index.html.

[11] A. Miles, M. Loughlin, and A. Polychronis. Medicine and evidence: knowledge and action in clinical practice. *Evaluation in Clinical Practice*, 13(4):481–503, August 2007.

[12] K. O'Hara and A. Sellen. A comparison of reading paper and on-line documents. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 335–342, New York, NY, USA, 1997. ACM.

[13] M. Peleg, S. W. 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. Comparing Computer-Interpretable Guideline Models: A Case-Study Approach. *Journal of the American Medical Informatics Association (JAMIA)*, 10(1):52–68, Jan-Feb 2003.

[14] K.-A. Polvani, A. Agrawal, B. Karras, A. Deshpande, and R. Shiffman. *GEM Cutter Manual.* Yale Center for Medical Informatics, 2000.

[15] M. Růžička and V. Svátek. Mark-up Based Analysis of Narrative Guidelines with the Stepper Tool. In K. Kaiser, S. Miksch, and S. W. Tu, editors, *Computer-based Support for Clinical Guidelines and Protocols. Proceedings of the Symposium on Computerized Guidelines and Protocols (CGP 2004)*, volume 101 of *Studies in Health Technology and Informatics*, pages 132–136, Prague, Czech Republic, 2004. IOS Press.

[16] Y. Shahar, O. Young, E. Shalom, A. Mayaffit, R. Moskovitch, A. Hessing, and M. Galperin. DEGEL: A hybrid, multiple-ontology framework for specification and retrieval of clinical guidelines. In M. Dojat, E. Keravnou, and P. Barahona, editors, *Proceedings of the 9th Conference on Artificial Intelligence in Medicine in Europe, AIME 2003*, volume 2780 of *LNAI*, pages 122–131, Protaras, Cyprus, 2003. Springer Verlag.

[17] R. Steele and J. Fox. Tallis PROforma Primer – Introduction to PROforma Language and Software with Worked Examples. Technical report, Advanced Computation Laboratory, Cancer Research, London, UK, 2002.

[18] P. Votruba, S. Miksch, and R. Kosara. Facilitating knowledge maintenance of clinical guidelines and protocols. In M. Fieschi, E. Coiera, and Y.-C. J. Li, editors, *Proceedings from the Medinfo 2004 World Congress on Medical Informatics*, pages 57–61. IOS Press, 2004.

[19] S. H. Woolf. Evidence-based medicine and practice guidelines: an overview. In *Cancer Control 7*, 4, pages 362–367, 2000.