Research Paper ■

A Pattern-based Analysis of Clinical Computer-interpretable Guideline Modeling Languages

NATALIYA MULYAR, MSC, WIL M. P. VAN DER AALST, PHD, MOR PELEG, PHD

Abstract **Objectives:** Languages used to specify computer-interpretable guidelines (CIGs) differ in their approaches to addressing particular modeling challenges. The main goals of this article are: (1) to examine the expressive power of CIG modeling languages, and (2) to define the differences, from the control-flow perspective, between process languages in workflow management systems and modeling languages used to design clinical guidelines.

Design: The pattern-based analysis was applied to guideline modeling languages Asbru, EON, GLIF, and PRO*forma*. We focused on control-flow and left other perspectives out of consideration.

Measurements: We evaluated the selected CIG modeling languages and identified their degree of support of 43 control-flow patterns. We used a set of explicitly defined evaluation criteria to determine whether each pattern is supported directly, indirectly, or not at all.

Results: PRO*forma* offers direct support for 22 of 43 patterns, Asbru 20, GLIF 17, and EON 11. All four directly support basic control-flow patterns, cancellation patterns, and some advance branching and synchronization patterns. None support multiple instances patterns. They offer varying levels of support for synchronizing merge patterns and state-based patterns. Some support a few scenarios not covered by the 43 control-flow patterns.

Conclusion: CIG modeling languages are remarkably close to traditional workflow languages from the control-flow perspective, but cover many fewer workflow patterns. CIG languages offer some flexibility that supports modeling of complex decisions and provide ways for modeling some decisions not covered by workflow management systems. Workflow management systems may be suitable for clinical guideline applications.

■ **J Am Med Inform Assoc.** 2007;14:781–787. DOI 10.1197/jamia.M2389.

Introduction

Clinical practice guidelines and protocols apply to diverse areas, including policy development, utilization management, education, reference, clinical decision support, conduct of clinical trials, and workflow facilitation. Clinical guidelines seek to improve the quality of patient care and reduce costs. Creating computer-interpretable representations of the clinical knowledge supporting clinical guidelines is crucial for developing decision-support systems that provide patient-specific advice at the point of care. Automated guideline-based systems can improve adherence over paper-based guidelines.[1]

Affiliations of the authors: Department of Eindhoven University of Technology (NM, WMPV), Eindhoven, the Netherlands, Department of Management Information Systems, University of Haifa (MP), Haifa, Israel.

The authors thank Samson Tu, Silvia Miksch, Andreas Seyfang, and Paolo Ciccarese for their contributions in the tool evaluations; Richard Thomson, Nick J. Wells, and John Fox for providing an access to the Tallis tool set and for the cross-check of the evaluation results; and Nick Russell for the input he provided to the revision of the control-flow patterns.

Correspondence: Nataliya Mulyar, MSc, Eindhoven University of Technology Paviljoen J.08, NL-5600 MB Eindhoven, the Netherlands; e-mail: <n.mulyar@tue.nl>.

Received for review: 01/30/07; accepted for publication: 07/26/07.

Although many parties have been engaged in developing languages for representing computer-interpretable guidelines (CIGs),[2–9] little standardization exists of languages that fully support representation of the logic of guidelines that unfold over time. Standards would facilitate sharing and enable adaptation in local practice settings.[10] Indeed, the three standards, Arden Syntax, Guidelines Elements Model (GEM), and Guideline Expression Language, Object-oriented (GELLO), which have been developed in the domain of clinical decision support, do not satisfy these requirements. GEM,[11] a standard of the American Society for Testing and Materials (ASTM), is an XML-based knowledge model for guideline documents. GEM elements relate to a guideline's identity, developer, purpose, intended audience, method of development, target population, knowledge components, testing, and review plan. Although this standard includes elements for marking up components of clinical algorithms, the resulting markup does not support computer execution that requires automatic inference. The Arden Syntax[12] is a standard of ASTM and of Health Level Seven (HL7) that has been substantially used in industry. This standard is suitable for representing individual decision rules in self-contained units called Medical Logic Modules (MLMs), which are usually implemented as event-driven alerts or reminders. Arden Syntax is not designed for encoding complex multistep guidelines that unfold over time and does not offer mechanisms for complexity management and for managing linked MLMs.[13] Guideline Expression Language, Object-oriented (GELLO),[14] which has been recently accepted as an HL7 and ANSI standard, is a vendor-independent, object-oriented, side-effect-free, and extensible expression language that could be used for specifying and sharing decision logic and eligibility criteria, calculations, patient state definitions, conditions, and system actions. Because it was developed as an expression language, it does not support specification of entire clinical algorithms, but focuses on specifying logical expressions. GELLO is the first component of a CIG language that HL7 started to standardize to support a full CIG formalism. The other components that HL7 sought to standardize include, among others, a control-flow language.[15]

Because there is no standard CIG formalism, this article concentrates on nonstandard CIG formalisms of the type termed task-network models (TNMs).[15] TNM CIG formalisms have in common a process-flow-like model that decomposes guidelines into a network of tasks that unfold over time, but they differ from each other in their approaches to addressing particular modeling challenges. Investigators[15] compared six guideline modeling languages: Asbru, EON, **G**uide**l**ine **I**nterchange **F**ormat (GLIF), GUIDE, Prescribing Rationally with Decision-support In General-practice studY (PRODIGY), and PRO*forma* according to eight components that capture the structure of CIG languages (see Related Work section). In this article, we examine the modeling languages using control-flow patterns. The control-flow patterns have been tested by evaluating a multitude of workflow systems and standards. The feedback from industry has resulted in the revision and extension of the control-flow patterns, which nowadays serve as an accepted benchmark.[17] The evaluation of CIG modeling languages is a big challenge because the terminology used in these languages is inconsistent, the semantics of the control-flow of some of the languages is incompletely and informally defined, and the approaches used by the languages for guideline modeling are heterogeneous. CIGs represent clinical algorithms that unfold over time by specifying the ordering of tasks and activities. The ordering of tasks in a process model is also referred to in the literature as control-flow, which is the perspective we focused on during the analysis. We compared the control-flow component of CIG languages by evaluating their degree of support of control-flow patterns[16,17] that are known as workflow patterns. Although workflow patterns come from the business process-modeling community, they are suitable for comparing CIG languages. A CIG language is a computer-interpretable TNM of a clinical care process that realizes a clinical/medical goal, while a workflow model is a computer-interpretable TNM of a business process that realizes a business objective. From the control-flow perspective, both of these types of models (languages) are TNMs and are comparable from the control-flow perspective.

Initially, we intended to analyze the current versions of the same set of TNM languages as considered in Peleg[15]: Asbru,[31] EON,[3] GLIF,[4] NewGuide,[7] PRODIGY[5] and PRO*forma*.[39] However, we excluded from our analysis NewGuide because it is still under development and PRODIGY because it is no longer actively supported.

Background

This section describes the main concepts of the CIG modeling languages Asbru, EON, GLIF, and PRO*forma* and presents work related to workflow patterns.

Computer-interpretable Guidelines

Table 1 illustrates terms used in the CIG modeling languages that correspond to the main workflow concepts that will be used throughout this article. These terms include process model, case, task, parallel branching, and exclusive branching and are defined in van der Aalst et al.[18] A process model consists of a number of tasks that have to be carried out and a set of conditions that determine the order of tasks. A task is a logical unit of work that is carried out as a whole. Tasks can be executed based on sequential, parallel or conditional routing. Parallel branching specifies that two or more tasks are executed independently of each other. Exclusive branching splits a process in several branches, only one of which can be selected based on the fulfillment of a condition associated with a given branch.[19] Process models are executed for specific cases (e.g., a patient with high blood pressure being managed by a hypertension CIG). Each case involves a process being performed, with its current active tasks. In Appendix 1 (available as a JAMIA online data supplement at www.jamia.org), we describe in more detail the main concepts of CIG modeling languages by modeling a patient diagnosis scenario in Asbru/AsbruView, EON/Protege-2000, GLIF/Protege-2000, and PROforma/Tallis, as shown in Figures 1-4 respectively, in the online data supplement.[29]

Workflow Patterns

The recent Workflow Patterns Initiative[17] has taken an empirical approach to identifying the most common control constructs inherent to modeling languages adopted by workflow systems. In particular, a broad survey of modeling languages resulted in 20 workflow patterns being identified.[20] The collection of patterns was originally limited to the control-flow perspective, thus the data, organizational, and application perspectives were missing. In addition, the set of control-flow patterns was not complete because the patterns were gathered nonsystematically: they have been obtained as a result of an empirical analysis of the modeling facilities offered by selected workflow systems. The first shortcoming has been addressed by means of the systematic analysis of data and resource

Table 1 ■ Terms Used by Asbru, EON, GLIF, and PRO*forma*

Terms	Asbru	EON	GLIF	PRO*forma*
Process model	Plan	Guideline	Guideline	Plan
Case	Instance of plan	Guideline Instance	Guideline Instance	Instance of plan
Task/activity	Plan	Action	Action	Action, enquiry
Parallel branching	Plan type	Branch and synchronization	Branch and synchronization	Action or enquiry
Exclusive branching	Plan precondition, plan type	Decision	Decision	Decision, enquiry and scheduling constraints

Table 2 ■ Support for the Control–flow Patterns in Asbru, EON, GLIF, and PRO*forma*

	Asbru	EON	GLIF	PRO*forma*
Basic control-flow				
1. Sequence	+	+	+	+
2. Parallel split	+	+	+	+
3. Synchronization	+	+	+	+
4. Exclusive choice	+	+	+	+
5. Simple merge	+	+	+	+
Advanced branching and synchronization				
6. Multichoice	+	+	+	+
7. Structured synchronizing merge	+/−	−	−	+
8. Multimerge	−	−	−	−
9. Structured discriminator	+	+	+	+
Structural patterns				
10. Arbitrary cycles	−	+	+	−
11. Implicit termination	+	+	+	+
Multiple instances patterns				
12. MI without synchronization	−	−	−	−
13. MI with a priori design-time knowledge	+/−	+/−	+/−	+/−
14. MI with a priori run-time knowledge	−	−	−	−
15. MI without a priori run-time knowledge	−	−	−	−
State-based patterns				
16. Deferred choice	+	−	+	+
17. Interleaved parallel routing	+	−	−	−
18. Milestone	−	−	−	+
Cancellation patterns				
19. Cancel activity	+	+	+	+
20. Cancel case	+	−	+/−	+
New patterns				
21. Structured loop	+	+	+	+
22. Recursion	+	−	−	−
23. Transient trigger	−	−	−	+
24. Persistent trigger	−	−	+	+
25. Cancel region	−	−	−	−
26. Cancel multiple instance activity	+	−	+	+
27. Complete multiple instance activity	+	−	−	+
28. Blocking discriminator	−	−	−	−
29. Canceling discriminator	+	−	−	+
30. Structured N-out-of-M join	+	−	+	+
31. Blocking N-out-of-M join	−	−	−	−
32. Canceling N-out-of-M join	−	−	−	+
33. Generalized AND-join	−	−	−	−
34. Static N-out-of-M join for MIs	−	−	−	−
35. Static N-out-of-M join for MIs with cancellation	−	−	−	−
36. Dynamic N-out-of-M join for MIs	−	−	−	−
37. Acyclic synchronizing merge	−	−	−	+
38. General synchronizing merge	−	−	−	−
39. Critical section	+	−	+	−
40. Interleaved routing	+	−	+	−
41. Thread merge	−	−	−	−
42. Thread split	−	−	−	−
43. Explicit termination	−	−	−	−

(+) full support; (+/−) partial support; (−) no support.
MI = multiple instances.

perspectives and resulted in the extension of the collection of the control-flow patterns by 40 data patterns and 43 resource patterns.[21,22] The issue of the incompleteness of the control-flow patterns has been resolved by means of the systematic analysis of the classical control-flow patterns against Workflow Pattern Specification Language.[23] Furthermore, the originally identified set of the 20 control-flow patterns has been revised and extended with 23 new patterns. A comprehensive description of the full set of 43 control-flow patterns is found in Russell et al.[16]

The 43 patterns can be divided into several groups: basic control-flow patterns, advanced branching and synchronization patterns, structural patterns, multiple instances patterns, state-based patterns, cancellation patterns, and the 23 new patterns that will be classified outside the scope of this research. Due to the lack of space, in this article we provide only the description of patterns that have received different ratings by the examined languages, and are therefore the most interesting. These definitions are given in the Results

Table 3 ■ Description of Pattern Categories

Category name	Description
Basic control-flow patterns	Patterns describing elementary aspects of process control: sequence, parallel split, synchronization, exclusive choice, and simple merge
Advanced branching and synchronization	Patterns describing in-between behaviors, where some of the paths in a set of paths can be selected for execution and different modes of continuation are possible thereafter
Structural patterns	Structural patterns identify whether the modeling formalism has any restrictions regarding the structure of the processes
Multiple instances patterns	Patterns that refer to situations where several instances of a task can be active concurrently in the same case
State-based patterns	Patterns characterizing scenarios in a process where subsequent execution is determined by the state of the process instance
Cancellation patterns	Patterns refer to the situation where either a single task or a group of tasks have to be cancelled in a model
New patterns	A set of new patterns and the revised variants of patterns in the above-introduced categories that address the concepts such as triggers, path and thread branching and synchronization, and cancellation

section, so that the discussion of the different ways in which the CIG languages support these patterns could be easily followed.

Workflow patterns have become a standard for assessing strengths and weaknesses of process specifications. Many workflow systems and standards such as XML Processing Definition language (XPDL), Unified Modelling Language (UML), Business Process Execution Language (BPEL), The eXtensible Language (XLANG), Web Services Flow Language (WSFL), Business Process Modelling Language (BPML), and Web Service Choreography Interface (WSCI) were evaluated from the perspective of the control-flow patterns, a summary of which is available[17] The patterns have inspired the improvement and development of 10 languages and tools.[17] Furthermore, the workflow patterns were used for selecting a workflow management system (WfMS) (i.e., a system in which workflows are defined, created, and executed) and have been used in teaching.[17]

Research Questions

The main research questions addressed by this study are: "What is the degree of support of the control-flow patterns in special-purpose languages for modeling clinical guidelines?" and "What are the differences, from the control-flow perspective, between process languages offered by workflow management systems and modeling languages used to design clinical guidelines?"

Methods

In this section we describe the types of analyses that we carried out and the criteria used for evaluating the pattern support offered by the examined CIG modeling languages.

Analysis

We evaluated the set of CIG languages against the revised set of 43 control-flow patterns, described in detail.[16]

To compare the examined languages, we used quantitative and qualitative measures. We calculated the number of patterns supported by the examined languages directly, indirectly, and not supported at all. Furthermore, we analyzed in greater detail the differences between the languages based on the support of patterns that have received different ratings. In particular, we underlined the strengths of CIG languages that were unique in their support of particular patterns, the significance of this support to clinical guidelines, the different ways in which the considered languages support the workflow patterns, and how they differ from process modeling languages used in the business domain.

Evaluation Criteria

For each language, we checked whether it is possible to realize the control-flow pattern with the facilities offered by the language. As a means for evaluation, we used evaluation criteria explicitly defined.[16] These evaluation criteria specify a set of context conditions an analyzed language has to fulfill in order to support a pattern. The pattern support has been rated as full, partial, or no support. A pattern is fully supported (+) if the examined language fully satisfies the evaluation criteria for the pattern and provides direct support for each of them via constructs found in the language. A pattern is supported partially (+/−) if the examined language provides indirect support for all of the criteria either via extended workarounds or programmatic extensions. A pattern is not supported (−) if the examined language does not satisfy any of the criteria for direct or indirect support. To make sure that our understanding of the CIG languages abilities was correct, the developers of the four languages that we compared reviewed our article before its submission.

Results

Comparing CIG Languages Support of Categories of Workflow Patterns

Table 2 summarizes the support of the full set of 43 patterns by the languages. The brief description of pattern categories used for the evaluation is given in Table 3. We explicitly elaborate on patterns that received different ratings by the examined languages (i.e., patterns that are supported only by a subset of the examined languages), which underline the weaknesses and strengths of these languages essential for understanding of the article in Appendix 2 (available as a JAMIA online data supplement at www.jamia.org). We provide the full set of results in an online source.[24]

After analyzing how the four CIG languages support the specific workflow patterns, as summarized in Table 2, we tried to arrive at more general conclusions about the languages' support of categories of workflow patterns. As the results of the analysis have shown, PRO*forma* offers direct support for the largest number of patterns (22 of 43) among the examined offerings. Asbru and GLIF offer support for 20 and 17 patterns, respectively. Even fewer patterns are supported by EON (it supports only 11 patterns).

More detailed analysis of the pattern support reveals that all examined offerings directly support basic control-flow patterns. At least half of the advanced branching and synchronization patterns, which are relatively common to business processes used in practice, are supported by all offerings. Note that the structured synchronizing merge pattern is not supported by all examined offerings. Although PRO*forma* supports this pattern directly, Asbru adds a time restriction to the process of synchronization to approximate the desired behavior. The semantics of the synchronization blocks in EON and GLIF are not precise enough, i.e., they do not specify what happens to the active tasks after the synchronization task has been executed. This also is the reason why some of the new patterns addressing variants of the synchronization merge are not supported by EON and GLIF.

None of the examined modeling languages have the concept of a multiple instance activity, and therefore, patterns from the multiple instances pattern group and new patterns related to the multiple instances activity are not supported directly.

Not all examined languages have full support for the state-based patterns. Although EON and GLIF have the notion of the patient state, they lack the notion of the process state. The only language that used these concepts is PRO*forma*. All analyzed languages support the cancellation patterns relatively well.

Unique Features of the CIG Languages

While evaluating the modeling languages and studying their documentation, we identified several scenarios not covered by the set of the control-flow patterns that we had used as a reference framework. In particular, a deferred multichoice is a capability to defer the selection of multiple options by a user until the user decides that no more options will be selected (for instance, selecting several medicines from the recommended ones for the treatment of the patient). The functionality of the deferred multichoice has been encountered in GLIF3.5/Protege-2000, EON/Protege-2000 and PRO*forma*/Tallis. Another scenario is related to forced trigger, where any internal or external event triggers the execution of a task even if the task precondition was not satisfied at the moment of triggering. The functionality of the forced trigger has been encountered in PRO*forma*/Tallis.

In addition, guideline modeling languages allow for some flexibility by offering expression languages that support modeling of complex decisions. They also provide ways for modeling decisions as argumentation rules (rule-in and rule-out), which are unique features that affect control-flow specification and are not offered by workflow management systems.

Another aspect of flexibility, offered in EON and GLIF, is the ability to specify multiple entry and exit points to a guideline. Such a feature might be useful when, due to unpredictable changes in a patient's state, a patient has to jump from one state of the guideline, at which he was situated at the previous encounter, to another state that reflects his current situation (e.g., his condition deteriorated despite the use of the guideline, or due to a different guideline that was applied to him, medications were added, etc.). However, such support of multiple entry points is not unique to EON and GLIF and has alternatives; similar behavior can be achieved by means of the state triggers in PRO*forma*.

Discussion

Members of the computerized guidelines community have emphasized how important it is to support flexibility in guideline formalisms.[4,15,25] However, when we examined guideline modeling languages, we found only limited additional flexibility not present in business process modeling languages. The CIG languages we studied support only two new patterns not encountered in business process models. This is remarkable because one would have expected dedicated constraints allowing for more flexibility given the more dynamic nature of care processes.

Moreover, only half of the workflow patterns elicited from business process modeling languages are supported by CIG languages. An interesting question is whether the patterns that are not supported by CIG languages could be useful in the domain of clinical guidelines automation. Many of these patterns relate to flexibility of process execution. In the business processes domain, multiple threads of execution that relate to the same activity are often supported (e.g., an insurance claim with a variable number of witness statements or an order containing multiple order lines). Similar situations may arise when a clinical trial is executed for groups of patients, for example. To identify whether there is a need for CIG modeling constructs supporting multiple instances, more research has to be done addressing the nature of the clinical guidelines requirements.

Because CIG languages do not offer substantially more control-flow constructs than business process modeling languages, the medical community might rethink the use of more general formalisms and tools, which have formal foundation and have been widely tested and used in industry, for expressing control flow of guideline models. For instance, the case-handling system FLOWer[26] offers a high degree of flexibility during the execution of a case (i.e., a process instance). FLOWer is based on an information-driven approach and takes the process as its focal point, whereas traditional workflow management systems are based on the routing of activities from work tray to work tray, leading to inflexibility. Although FLOWer suggests which steps have to be performed according to the modeled process description, a user is able to execute any task from the given list, even to re-execute some of them. This may be very useful for clinicians who are using guidelines and disagree with the advice provided by the CIG because they think that their patient's case was not considered by the developers of the CIG or that new evidence suggests another treatment option. We note that some of the CIG execution engines (e.g., GLIF's execution engine Guideline Expression Language Object-oriented (GLEE)) support execution of any task that is defined in the CIG, at any point in time, if the user wishes to do so. Yet, this execution semantics is not part of the semantics defined for the GLIF language.

In addition to the set of constructs discussed in this article, the medical community may also consider using configurable modeling constructs, found in business process formalisms.[27] A CIG developed by one organization can be locally adjusted by another organization by using configurable modeling constructs. Such configurable constructs enable specifying ahead of time what part of a model can be configured and how. For instance, a choice between various kinds of tests performed by a laboratory can be configured to a choice between a blood analysis and a urine analysis that are performed by an assistant of a family doctor. This is very important, as some changes that are made locally could violate the purpose of the guideline and it is therefore important to define what changes should be permitted.

Another area that has been developing in the business process community and could benefit the CIG community (especially if it would adopt a workflow-based semantics of process models) is the area of process mining.[28] Mining logs of executed events (e.g., medication ordering, patient referrals) can be used to discover the actual workflow of patient care and how it deviates from a CIGs process model.

The results of the evaluation presented in this article could be used to clarify language specifications. Moreover, the evaluation results can be used as a means for comparing the capabilities of the languages to express the control-flow patterns and for selecting an appropriate modeling language. For instance, medical organizations that plan to automate their processes and improve the quality of care by using CIGs may match the list of their requirements against the results of the pattern-based evaluation. For example, if an organization requires exclusive execution of activities in non-predefined order, then Asbru might be chosen, because no other language from the evaluated ones offers these feature (see pattern 17). If a requirement is to incorporate transient triggers (pattern 23), then the best choice would be PRO*forma*; persistent triggers (pattern 24) also are supported by GLIF. PRO*forma* is also a good choice if such requirements as synchronization of variable number of paths (pattern 37) or support of milestones (pattern 18) are important. The milestone pattern is important for modeling medical guidelines. For example, in a cancer protocol, two treatment strategies could be used: a surgery or medication. A surgery may be performed only if medication cannot be prescribed or it does not help. Checking the state of medication effect before enabling the surgery could be done by means of the milestone pattern. GLIF or EON could be a language of choice if flexibility in the structure of a guideline is required (they support the arbitrary cycles pattern).

The analysis we performed and reported in this article has several limitations. It concentrates only on the control-flow aspect of the guideline formalisms and does not take into consideration other aspects such as data and resources. Furthermore, the evaluation has been performed on the limited set of the languages. In particular, a few formalisms that are recognized as standards, e.g., Arden syntax and GEM, were not included in the study. Note that Arden syntax has been excluded because it is used to model individual decisions (not guidelines that unfold over time). GEM is focused on the guideline DOCUMENT model—structuring the evidence statements and the decision variable. GEM permits to markup text as imperative recommendations or as parts of decisions tables; at the same time, it misses the logic of a guideline that unfolds over time.

Conclusion

From a flow-control perspective, the Asbru, EON, GLIF3.5, and PRO*forma* CIG languages are very similar to the process languages of workflow management systems, although they do not make use of many of the workflow patterns in such systems. The additional workflow patterns supported by process languages of workflow management systems may be useful for clinical guideline applications. A suitable CIG can be selected for a specific modeling and execution task on the basis of pattern-based requirements.

References ■

Note: References 30, 32–38, and 40 are cited in the online data supplement to this article at jamia.org.

1. Overhage JM, Tierney WM, Zhou XH, McDonald CJ. A randomized trial of "corollary orders: to prevent errors of omission. J Am Med Inform Assoc 1997;4:364–75.
2. Votruba P, Miksch S, Kosara R. Facilitating knowledge maintenance of clinical guidelines and protocols. Medinfo 2004; 11:57–61.
3. Tu SW. The EON guideline model. Technical report. 2006. Available at: http://smi.stanford.edu/projects/eon/EONGuidelineModelDocumentation.doc. Accessed May 2, 2007.
4. Boxwala AA, Peleg M, Tu S, et al. GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines. Biomed Inform 2004;37:147–61.
5. Johnson PD, Tu SW, Booth N, Sugden B, Purves IN. Design and implementation of a framework to support the development of clinical guidelines. Proc AMIA Symp 2000:389–93.
6. de Clercq PA, Hasman A, Blom JA, Korsten HH. Design and implementation of a framework to support the development of clinical guidelines. Int J Med Info 2001;64:285–318.
7. Ciccarese P, Caffi E, Quaglini S, Stefanelli M. Architectures and tools for innovative Health Information Systems: The Guide Project. Int J Med Info 2005;74:7–8.
8. Tu S, Musen M. A flexible approach to guideline modeling. AMIA Symp 1999:420–4.
9. Wang D, Peleg M, Tu S, et al. Representation primitives, process models and patient data in computer-interpretable clinical practice guidelines: a literature review of guideline representation models. Int J Med Inform 2002;59–70.
10. Peleg M. Guideline and workflow models. In: Greenes RA (ed). Medical Decision Support: Computer-Based Approaches to Improving Healthcare Quality and Safety. New York: Elsevier, 2006, pp. 281–306.
11. Shiffman RN, Karras BT, Agrawal A, Chen R, Marenco L, Nath S. GEM: a proposal for a more comprehensive guideline document model using XML. J Am Med Inform Assoc 2000;7:488–98.
12. Hripcsak G, Ludemann P, Pryor TA, Wigertz OB, Clayton PD. Rationale for the Arden Syntax. Comput Biomed Res 1994;27: 291–324.
13. Peleg M, Boxwala AA, Bernstam E, Tu S, Greenes RA, Shortliffe EH. Sharable representation of clinical guidelines in GLIF: relationship to the Arden Syntax. J Biomed Inform 2001;34:170–81.
14. Sordo M, Ogunyemi O, Boxwala AA, Greenes RA, Tu S. GELLO: A Common Expression Language. Available at: http://dsg.bwh.harvard.edu/~sordo/GELLO/GELLO-073105short.htm. Accessed September 28, 2007.
15. Peleg M, Tu SW, Bury J, et al. Comparing computer-interpretable guideline models: a case-study approach. J Am Med Inform Assoc 2003;10:52–68.

16. Russell N, ter Hofstede AHM, van der Aalst WMP, Mulyar N. Workflow Control-Flow Patterns: A Revised View. BPM Center Report BPM-06-22, BPMcenter.org. 2006. Available at: http://workflowpatterns.com/documentation/documents/BPM-06-22.pdf. Accessed April 30, 2007.
17. Workflow Patterns Home Page. Available at: http://www.workflowpatterns.com. Accessed April 30, 2007.
18. van der Aalst W, van Hee K. Workflow management. Cambridge, MA: MIT Press, 2004.
19. Dumas M, van der Aalst W, ter Hofstede AH. Process-Aware Information Systems: Bridging People and Software Through Process Technology. New York: Wiley, 2005.
20. Kiepuszewski B. Expressiveness and Suitability of Languages for Control Flow Modelling in Workflows. PhD thesis, Queensland University of Technology, Brisbane, Australia, 2003. Available at: http://www.workflowpatterns.com/documentation/documents/phd_bartek.pdf. Accessed September 28, 2007.
21. Russell N, ter Hofstede AHM, Edmond D, van der Aalst WMP. Workflow Resource Patterns: Identification, Representation and Tool Support, *CAISE 2005*. Portugal: Porto, pp. 216–32.
22. Russell N, ter Hofstede AHM, Edmond D, van der Aalst WMP. Workflow Data Patterns. QUT Technical report, FIT-TR-2004-01. Brisbane: Queensland University of Technology, 2004.
23. Mulyar N, van der Aalst WMP, ter Hofstede AHM, Russell N. Towards a WPSL: A Critical Analysis of the 20 Classical Workflow Control-flow Patterns. Technical report, BPM Center Report BPM-06-18. 2006. Available at: http://is.tm.tue.nl/staff/wvdaalst/BPMcenter/reports/2006/BPM-06-18.pdf. Accessed September 28, 2007.
24. Mulyar N, van der Aalst WMP, Peleg M. A Pattern-based Analysis of Clinical Computer-Interpretable Guideline Modeling Languages. BPM Center Report BPM-06-29. 2006. Available at: http://is.tm.tue.nl/staff/wvdaalst/BPMcenter/reports/2006/BPM-06-29.pdf. Accessed September 28, 2007.
25. Quaglini S, Stefanelli M, Lanzola G, Caporusso V, Panzarasa S. Flexible guideline-based patient careflow systems. Artif Intell Med 2001;22:65–80.
26. Pallas Athena. Pallas Athena website. Available at: http://www.pallasathena.nl. Accessed May 2, 2007.
27. van der Aalst WMP, Dreiling A, Gottschalk F, Rosemann M, Jansen-Vullers MH. Configurable Process Models as a Basis for Reference Modeling. Business Process Management Workshops 2005. LNCS. New York: Springer, 2005, pp. 512–18.
28. Rozinat A, Mans RS, van der Aalst WMP. Mining CPN models: discovering process models with data from event logs. In: Jensen K (ed). Proceedings of the Seventh Workshop on the Practical Use of Coloured Petri Nets and CPN Tools (CPN 2006). Aarhus, Denmark: University of Aarhus, 2006.
29. AsbruView. Available at: http://www.ifs.tuwien.ac.at/asgaard/asbru/tools.html. Accessed December 13, 2006.
30. DELT/A. Document Exploration and Linking Tool/Addons. Available at: http://ieg.ifs.tuwien.ac.at/projects/delta. Accessed May 2, 2007.
31. Shahar Y, Miksch S, Johnson P. The Asgaard Project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines. Artif Intell Med 1998;14:29–51.
32. Kosara R, Miksch S. Metaphors of movement: a visualization and user interface for time-oriented skeletal plans. Artif Intell Med 2001;22:111–31.
33. CareVis. Available at: http://ieg.ifs.tuwien.ac.at/projects/carevis. Accessed April 30, 2007.
34. Aigner W, Miksch S. CareVis: integrated visualization of computerized protocols and temporal patient data. Artif Intell Med 2006;37:203–18.
35. Seyfang A, Kosara R, Miksch S. Asbru Reference Manual, Version 7.3. Technical Report. Vienna: Vienna University of Technology, Institute of Software Technology, 2002. Report No.: Asgaard-TR-2002-1, 2002.
36. Tu S, Musen M. Representation Formalisms and Computational Methods for Modeling Guideline-Based Patient Care. In: First European Workshop on Computer-based Support for Clinical Guidelines and Protocols. Leipzig, Germany: 2000, pp. 125–42.
37. Tu SW, Musen MA. A flexible approach to guideline modeling. Proc AMIA Symp 1999:420–4.
38. Tu SW, Musen MA. From guideline Modeling to guideline execution: defining guideline-based decision-support services. Proc AMIA Annu Symp 2000:863–7.
39. Fox J, Johns N, Rahmanzadeh A. Disseminating medical knowledge: the PROforma approach. Artif Intell Med 1998;14:157–82.
40. Clercq PA, Blom JA, Korsten HH, Hasman A. Approaches for creating computer interpretable guidelines that facilitate decision support. Artif Intell Med 2004;31:1–27.