Methodological Review

Computer-interpretable clinical guidelines: A methodological review

CrossMark

Mor Peleg *

Department of Information Systems, University of Haifa, Haifa 31905, Israel

ABSTRACT

Clinical practice guidelines (CPGs) aim to improve the quality of care, reduce unjustified practice variations and reduce healthcare costs. In order for them to be effective, clinical guidelines need to be integrated with the care flow and provide patient-specific advice when and where needed. Hence, their formalization as computer-interpretable guidelines (CIGs) makes it possible to develop CIG-based decision-support systems (DSSs), which have a better chance of impacting clinician behavior than narrative guidelines. This paper reviews the literature on CIG-related methodologies since the inception of CIGs, while focusing and drawing themes for classifying CIG research from CIG-related publications in the Journal of Biomedical Informatics (JBI). The themes span the entire life-cycle of CIG development and include: knowledge acquisition and specification for improved CIG design, including (1) CIG modeling languages and (2) CIG acquisition and specification methodologies, (3) integration of CIGs with electronic health records (EHRs) and organizational workflow, (4) CIG validation and verification, (5) CIG execution engines and supportive tools, (6) exception handling in CIGs, (7) CIG maintenance, including analyzing clinician's compliance to CIG recommendations and CIG versioning and evolution, and finally (8) CIG sharing. I examine the temporal trends in CIG-related research and discuss additional themes that were not identified in JBI papers, including existing themes such as overcoming implementation barriers, modeling clinical goals, and temporal expressions, as well as futuristic themes, such as patient-centric CIGs and distributed CIGs.

© 2013 Elsevier Inc. All rights reserved.

1. Introduction

Trustworthy clinical practice guidelines (CPGs) are statements that include recommendations intended to optimize patient care that are informed by a systematic review of evidence and an assessment of the benefit and harms of alternative care options [1]. Clinical guidelines aim to improve the quality of care, limit unjustified practice variations and reduce healthcare costs. In order for them to be effective, clinical guidelines need to be integrated with the care flow, and provide appropriate recommendations when and where needed. Hence, their formalization as computer-interpretable guidelines (CIGs) [2] makes it possible to develop CIG-based decision-support systems (DSSs) through computer-based reasoning and execution of these formalized models. Such CIG-based DSSs match formalized guideline knowledge with updated patient clinical data to provide patient-specific advice at the point of care, increasing the chance of impacting clinician behavior compared to using only the narrative guidelines [3].

Research on CIGs started about 20 years ago and became more wide-spread in the late-1990s and early 2000s. Different approaches have been developed to represent and execute clinical guidelines over patient-specific clinical data. They include document-centric models, decision trees and probabilistic models, and "Task-Network Models" (TNMs) [2], which represent guideline knowledge in hierarchical structures containing networks of clinical actions and decisions that unfold over time. This review does not concern modeling of CPGs using case-based reasoning, chaining of individual decision rules (e.g., Arden syntax [4] or rule-based systems), or using CIG formalisms to represent individual decision rules (e.g., rules for adverse drug event prevention [5]). Both formal and semi-formal CIG representations are included in this review; formal models represent decision criteria in a formal interpretable language that enables binding of individual patient data item values to determine the value of the decision criteria, whereas semi-formal representation structures the clinical guideline as linked CIG elements where decision criteria can be represented informally, either as text or structured, but without an interpreter that can bind decision variables to values of patient data and evaluate complex criteria. Semi-formal CIGs could still be interpreted with user assistance (e.g., by making selections between decision options) and they have been used as intermediate step to simplify development of formal models (e.g., the Many-headed Bridge ap-

proach [6] which allows bridging the gap between knowledge contained in one or several narrative guidelines and one or several CIG formalisms).

In this paper, I review CIG-related methodologies since the inception of CIGs, while focusing and drawing themes for classifying CIG research from past JBI issues. Being a methodology-focused journal, this paper's hypothesis is that the papers related to this area published in JBI since 2001 would be representative of the different themes found in other medical informatics journals. Creating a manual classification of JBI papers was more tractable than categorizing the full set of papers from the five journals. While adding the remaining papers, I reviewed the themes that had been identified to see whether changes were needed.

The rest of this paper is organized as follows. Section 2 presents the methods used to construct this review. Section 3 presents the emerging themes identified through the methodology papers already published in JBI, ordering them in a life-cycle approach. Section 4 reviews the CIG-related literature in the five selected medical informatics journals, as well as older papers that presented important methods developed in the CIG field. The paper ends with a discussion of additional themes that were not identified in JBI papers, including existing and futuristic themes, trends identified in CIG research, and limitations of this review.

2. Methods

The starting point of this review was to collect JBI papers related to CIGs. The term "CIG" was not used in the search string as not all publications use this exact term; other terms include, among others, electronic clinical guidelines, computerized clinical guidelines. Therefore, I conducted broad Pubmed searches of JBI papers relating to clinical/medical guidelines ("Journal of Biomedical Informatics"[journal] clinical guidelines; "Journal of Biomedical Informatics"[journal] medical guidelines) and a second similar search for JBI papers related to clinical/care pathways (i.e., similar search strings containing the terms clinical pathway, clinical pathways, care pathway, or care pathways). The last search was conducted on February 7, 2013. Forty-three papers were retrieved from the search with the guideline terms and 16 from the pathway terms. The titles and abstracts of these papers were manually checked to prune out those papers that did not address CIGs (e.g., papers that addressed clinical DSSs that do not rely on a reusable guideline formalism) or papers that were not fundamentally about methodology, such as applications and their evaluation without stressing novel methodology, organizational approaches, comparative analyses (including reviews and lessons

learned), and the like. After this manual pruning step, 21 of the 59 collected papers remained, spanning the 12 years in which JBI has been published (2001–February 2013).

After reading the set of 21 papers, eight themes were identified. These were organized in a life-cycle approach, starting from knowledge acquisition and conceptualization and ending with CIG sharing.

Next, I followed the same Pubmed search and manual pruning procedure for papers published since 2001 in four additional prominent journals in the field of medical informatics: Journal of the American Medical Informatics Association (JAMIA), International Journal of Medical Informatics (IJMI), Artificial Intelligence in Medicine (AIIM), and Methods of Information in Medicine (MIIM). The numbers of papers found in each journal using the two search criteria are provided in Table 1. Because I decided to use a very wide search term (clinical guidelines, care pathways) so as not to miss CIG-related works in a specific journal searched, I had to limit the number of results and especially irrelevant results by focusing on the most prominent medical informatics journals.

When there have been papers since 2001 about a certain topic, earlier seminal papers on the same topic (either from journals or conference proceedings) were included if the later papers had a strong dependency on them. They are represented by the numbers in parentheses in Table 1.

3. Emerging themes in CIG research

After reviewing the 21 CIG papers from JBI, I identified eight themes. The themes span the entire life-cycle of CIG development, as shown in Fig. 1. The cycle begins with CPG analysis and CIG design. The usual practice is to base CIGs on previously published narrative CPGs. Hence the starting point of the life-cycle is knowledge acquisition and specification for improved CIG design. CIGs are defined using (1) CIG modeling languages while following (2) CIG acquisition and specification methodologies. To acquire CIGs, teams of knowledge engineers and clinical experts start with the CPGs, supplementing the knowledge contained in them with clarification of implicit medical knowledge and specifying this guideline knowledge in a CIG modeling languages using CIG authoring (or knowledge acquisition) tools. In order to provide patient-specific advice based on existing patient data from EHRs, the CIG can be (3) integrated with EHRs and organizational workflow. Such integration may necessitate restating of CIG logic according to availability of EHR data. Once the CIG is represented in a CIG language (i.e., at some point after step 2 is carried out), (4) validation and verification of the CIG can begin, to ensure that the CIG can be

Table 1
Number of CIG-related methodological papers found in prominent medical-informatics journals.

Journal	# Papers retrieved by PubMed search (added during review)		#of papers left after manual pruning		Total papers left since 2001
	Clinical guidelines, Medical guidelines	Clinical pathways, Care pathways	Guidelines	Pathways	
JBI	43	16	20	1{+1}[a]	21
JAMIA	74 (3)[b]	16	7 (3)	0	7
IJMI	95	11	12	1	13
AIIM	27 (1)	7	21 (1)	1{+1}	22
MIIM	46	3	5	0	5
Additional seminal papers[c]	Not applicable		5	0	0
Total	285 (4)	53	65 (9)	3	68

[a] Numbers in curly braces indicate number of papers in the pathways category that were also retrieved by searching for clinical guidelines and therefore were counted in the guidelines category.

[b] Numbers in parentheses show early seminal papers from the four other journals that later papers referred to.

[c] The numbers of additional seminal papers from other sources, not including the four journals, are provided in this row.
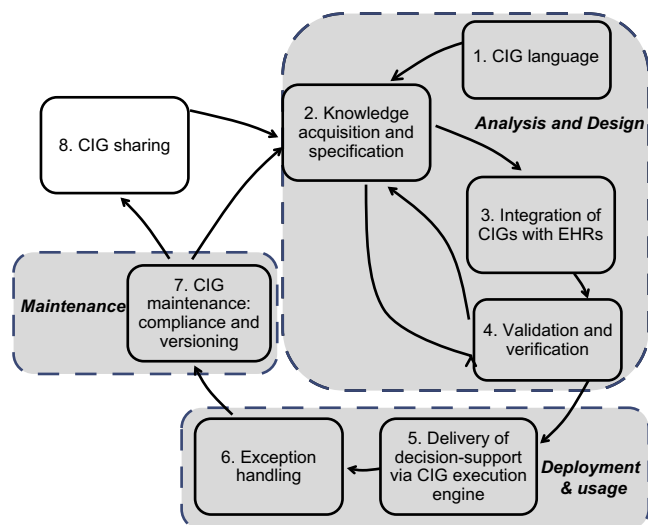
Fig. 1. Emerging topics in CIG research, arranged according to the CIG life-cycle.

safely applied to provide decision-support. When inconsistencies or errors are discovered in the CIG through validation and verification, CIG specification is revised and the CIG is validated and verified until it is safe for deployment. Hence a loop can be observed in Fig. 1 between the knowledge acquisition and specification step (step 2) and the validation and verification step (step 4). We distinguish between expert validation and formal verification, which are both supported by tools. Validation through testing is usually supported by (5) CIG execution engines, which are the tools used to deliver CIG-based decision-support.

During CIG execution for decision-support, exceptions may arise. Therefore, another topic of research is (6) exception handling in CIGs, which is an important facet in providing safe CIG-based DSS applications. With the aid of CIG execution engines, which can have exception-handling mechanisms, the validated and verified CIG-based DSS system can be deployed.

After the CIGs have been deployed and used, (7) the maintenance phase begins. In this phase two steps follow. First, to evaluate whether CIGs positively impact clinicians' behavior, clinicians' compliance to CIG recommendations should be measured and reasons for non-compliance should be studied. Second, as medical knowledge evolves, new diagnostic tests and therapeutic interventions become available and as evidence is collected about the outcome of CIG's recommendations and reasons for deviation from recommendations, CIGs should be evolved, leading to new versions. Once CIGs reach a stable state, (8) CIGs and CIG components could be shared with the community to increase reuse and save effort in developing new CIG-based DSS applications, although this is not always a design goal.

Table 2 lists the papers from the five journals and the additional seminal papers (including early papers from the 4 journals), classified into these eight themes.

4. Thematic review

In this section I review the papers that fit the selection criteria according to the eight identified themes, while referring to related work from the four other leading journals in the biomedical informatics field and to additional seminal papers published in other venues.

Table 2
The papers reviewed, arranged by theme.

Theme	Papers
(1) CIG modeling languages	JBI: 13, 23–25,27 JAMIA:30,31 IJMI: 8,9,11,17,18,29,31 AIIM:10,21,22,28 MIIM:19,26 Seminal: 7,15–16,33
(2) Knowledge acquisition and specification methodologies	JBI: 34–35,40 JAMIA:41,43 IJMI:36,44,50 AIIM:6,20,45,47,48,49 MIIM:39,46 Seminal: **7,16**,37,42
(3) CIG integration with EHRs and organizational workflow	JBI: **27**,ᵃ52 JAMIA:**31** IJMI:**29** AIIM:**28**,55 Seminal: 53
(4) CIG validation and verification	JBI: 58,64,69 JAMIA:60 AIIM:62,63,67,68,70 MIIM:65
(5) CIG execution engines and supportive tools	JBI: 73,75,78–79 IJMI:57,77 AIIM:76,80 Seminal: 74
(6) Exception handling in CIGs	JBI: 81,83 AIIM:82
(7) CIG maintenance: compliance and versioning	JBI: **58**,85,92,94 IJMI:90 AIIM:**47**,86,91,93 Seminal: 87
(8) CIG sharing	JBI: **13,23–25** JAMIA:95,97 Seminal: **33,53,74**

ᵃ Papers appearing second time in the table are shown in bold.

4.1. CIG languages

Different representations exist for CPGs, which can be partitioned into three main categories: document models, decision trees and probabilistic models, and task-network models.

4.1.1. Document models

The Guideline Elements Model (GEM) [7] is an XML-based knowledge model for guideline documents, developed by the research group of Shiffman. GEM's 110 elements relate to a guideline's identity, developer, purpose, intended audience, method of development, target population, knowledge components, testing, and review plan. Knowledge components in guideline documents include tags for marking term names and their definitions and are used to structure guideline recommendations as conditional recommendations (decision rules) and imperative recommendations (clinical actions). GEM's knowledge elements are deliberately less elaborated, as the knowledge component is the focus of task-network CIG models (see below) that could be integrated with GEM's document model. Recommendations can be sequenced using a link element to represent guidelines that unfold over time. GEM is a standard of the American Society for Testing and Materials (ASTM).

In [8], Georg et al. propose an extension of the GEM model that introduces additional levels of structuring centered on decision variables. They have demonstrated how they could automatically

generate a rule base from a textual guideline for the management of hypertension that has been marked-up as an instance of their extended GEM model.

4.1.2. Decision trees and probabilistic models

Several groups have been representing the guideline's algorithmic knowledge as a decision tree (DT). Generally speaking, a DT is a probabilistic model that can be evaluated by determining the expected utility or outcome and the optimal action strategy, i.e. the best strategy for every decision. However, DTs can also be used without probabilities to specify a hierarchical decision algorithm, as used in [9–11]. In [9], DTs are used as the conceptual organization of the guideline knowledge, independent from the encoding model. Each step of the DT is considered as an elementary knowledge which is to be explicated by elementary messages of prose text. To create a DT tree, the main sub problems addressed by the guideline are identified (e.g., screening, diagnosis, risk stratification, strategy of treatment). Next, the textual descriptions of possible decision conditions and corresponding elementary recommendations are identified. Based on these textual descriptions, the variables of decision and their possible values are formulated. Next, the hierarchy of variables is determined by identifying preliminary strict rule-in or rule-out conditions that should be tested before non-strict rules should be checked. Finally the decision tables (or matrix of variable values) for each identified sub-problems are created based on the rule hierarchy.

Decision trees are also the formalism used by Seroussi et al. [10]. Guideline knowledge is represented formally as decision trees. However, instead of automatically executing the decision tree, the user browses it as hypertext and flexibly interprets both patient data and guideline content, thus controlling the interpretation of the guideline knowledge in the specific context of a patient situation. In [11] the authors represent guideline-based therapeutic strategies as a two-level decision tree. The clinical level is used to identify a patient-specific clinical situation. The therapeutic level is derived from the formalization of guideline-based strategies first represented as bi-dimensional matrices structured in lines of therapy and levels of therapeutic intention. The derivation provides the most appropriate next step of treatment according to the patient's therapeutic history for the corresponding clinical situation.

Decision trees that are augmented with probabilities and utilities can be used for analyzing which decision options are most suitable for patients. Such DTs consider the uncertainty of the outcomes of selected therapies (decision options) by specifying the prior probabilities in the population for different outcomes (e.g., successful or unsuccessful treatment, risk of stroke as a side effect of treatment). They also consider the decision-maker's preferences by specifying her utilities for different outcome states, including possible adverse effects of treatment. DTs can be solved to provide the expected values of the payoffs for all the possible decision options from which personalized recommendations could be provided for particular populations and specific decision-makers. On the other hand, instead of specifying CPGs as decision trees, Sanders et al. [12] suggest that DTs could be the starting point for CPG development; CPG developers could construct DTs from population-based prior probabilities, analyze them to determine the expected values of payoffs and based on this analysis construct a CPG clinical algorithm that recommends optimal decision-making strategy.

Similar to decision trees, Fuzzy Cognitive Maps (FCMs) [13] can be used to represent and reason with causality and uncertainty. FCMs are networks consisting of nodes and connections. The nodes represent clinical concept variables, such as observations and ther-

apies and may take discrete or fuzzy values. The connections between the concepts and their strengths are drawn from clinical guidelines and are represented as if–then rules that denote how one concept is used to infer another concept (i.e., a therapy option could be linked to its indications and contra-indications and fuzzy weights of the connections, in the range of 0 and 1, would mark the strength of evidence). The authors demonstrated their approach by modeling part of a urinary tract infection guideline through FCMs using semantic web tools. When patient's data are input, the reasoning engine infers the values of activated nodes, ranking the different therapies. The modular representation of rules in FCMs could potentially facilitate knowledge reuse, sharing, and knowledge management.

4.1.3. Task-network models

The third category of representation is task-network models (TNMs) that represent CPGs formally allowing a CIG execution-engine to execute the represented knowledge against patient data, thus serving as DSSs that provides patient-specific recommendations at the point of care. TNMs hierarchically decompose clinical guideline algorithms (i.e., care processes) into networks of component tasks that unfold over time. Their computability is enabled by a formal specification of abstract concept as logical temporal expressions that are used in decision criteria. The procedural specification of clinical algorithms is termed "procedural knowledge" whereas the formal concept definitions that the procedural knowledge refers to is termed "declarative knowledge" [14]. A variety of task-network CIG languages have been developed over the years [2]. Most of the groups who have developed CIG languages have continued to publish methodological papers since 2001. These include the developers of GLIF, Guide, Asbru, GASTON, GLARE, HELEN, PROforma, and SAGE. The EON [15] CIG language is one of the exceptions, as its papers since 2001 describe only system evaluations rather than methodological papers, hence the original seminal EON paper is included in this review. A description of the emphasis of each of the task-network CIG languages is provided in Table 3, along with references to papers considered for this review. The summary below provides a short review of the challenges addressed by the different TNM formalisms [16–31].

All of the TNM formalisms have tools that support the visualization of clinical algorithms for easing human comprehension as well as a computable specification, for allowing automatic interpretation and execution. As was originally conceived in the deliberately simple task model of PROforma [30], first published in 1997, all of the TNMs support specification and execution of action, decision, data enquiry, and hierarchical plans and support parallel tasks. In GASTON [18–20] tasks can be specified as a set of problem-solving methods, such as task selection, monitoring tasks, preparation tasks and responding tasks. EON [15] uses a task-based approach to define decision-support services that can be implemented using alternative techniques. In contrast with PROforma, GLIF3 [25] is an extendible formalism; its object-oriented model lends itself to extension using the inheritance mechanism. Possibilities for extension include incorporation of different decision models, different expression languages (such as GEL and GELLO mentioned below), and different patient information models.

Decision-making logic supported by all TNM formalism allows representation of if-then-else rules. PROforma was also the first formalism to support argumentation logic, where rules in favor of and against taking a certain decision candidate are represented. This representation was also incorporated into EON and GLIF3.

Table 3
Task-network CIG formalisms and their emphasis.

CIG language	Focus	References
Asbru	In addition to specifying the guideline's action prescriptions, the process intentions (for care-provider actions) and outcome intentions (for patient states) of the guideline and of its major subplans are specified. Time-oriented actions, conditions, and intentions are expressed as patterns of actions or of external-world states that should be maintained, achieved, or avoided, during, or at the completion of a plan The second paper discusses how Asbru could be used to combine diagnosis and treatment plans. Treatment steps are modeled as a hierarchy of skeletal, time-oriented plans. Diagnosis can either be described in a declarative way in the conditions under which treatment steps are taken, or it can be modeled explicitly as plans of their own right	[16,17]
EON	EON's guideline model provides a core of an extensible set of models, such as a model for performing temporal abstractions and explanations. EON uses a task-based approach to define decision-support services that can be implemented using alternative techniques	[15]
GASTON	GASTON represents CIGs using primitives and ontologies to represent the medical domain and problem-solving methods (PSMs). The primitive classes are based on version 2.0 of GLIF. A guideline is associated with a task it has to solve. The task can be specified as a set of primitives and subguidelines or as an appropriate PSM, such as task selection, monitoring tasks, preparation tasks and responding tasks. Intention-based PSMs can be used to specify and execute GASTON guidelines at the intention-level	[18–20]
GLARE	GLARE emphasizes management of temporal knowledge. It includes comprehensive treatment of repetitions, periodicity and temporal constraints in clinical guidelines	[21–22]
GLIF3	GLIF emphasizes sharing of CIGs among implementing institutions and local adaptation of encoded guidelines (see also Section 4.8). It is supported by medical concept model, standard patient information model, and standard expression language for decision criteria. Apart from its visual flowchart level and the computable specification level, its implementable level of specification considers mapping CIG knowledge to EHR data	[23–25]
HELEN	Similar to GLIF3, HELEN's ontology of algorithm steps includes actions, decisions, branching, and nested subplans. Each algorithm strictly requires a Start_Step and each leaf of the algorithm must be terminated by a Diagnose step. In addition, Actions, Messages, or Decisions can be used. There are two types of decisions: Evaluate_by_users_choice or Evaluate_automatic. HELEN's expression language includes logical operators and arithmetic operators that can refer to variable values. Control flow structures include sequencing, conditional, synchronizing, wait statements, and iteration	[26]
NewGuide	NewGuide emphasizes careflow: fitting CIGs into organizational workflow, managing exceptions, and allowing the flexibility of deviating from the recommended flow when justified	[27–29]
PROforma	PROforma deliberately has a simple task model which is expressive enough to model guidelines yet easy enough to learn. Plans are networks of task subclasses: plan, action, enquiry, decision A particular property of decision tasks in PROforma is their *arguments*, which represent rules in favor of and against taking a certain decision candidate. The arguments are logical conditions that influence whether the candidate will be recommended. This is known as argumentation-based logic The paper cited here provides the syntax and semantics of PROforma, including semantics of PROforma expressions which may be used to query the state of a task during enactment and operational semantics that can be used for implementing public PROforma execution engines	[30]
SAGE	SAGE stands for Standards-Based Sharable Active Guideline Environment. A central concept of SAGE is context — characterized by a triggering event (e.g., patient checking in), patient characteristics, organizational setting (e.g., primary care outpatient clinic), organizational roles specifying who should respond to the event, and needed resources. Context coordinates the activation of guideline-based decision support. Procedural guideline logic is represented as an activity graph, which specifies how guideline-based clinical decision support should behave for a given scenario, and is assembled from steps that are based on the Workflow Management Coalition's process model. Recommendations that do not need to be organized and executed as part of a process are represented as decision maps. Like GLIF, SAGE also uses standard terminology, and HL7's Reference Information Model (RIM)	[31]

Time is a crucial concept for interpretation of medical data by identifying temporal patterns as well as for representation of time-oriented care plans, such as cyclical and periodical plans that are often found in cancer protocols. Representation and management of temporal aspects is emphasized in EON [15], Asbru [16,17], GLARE [20,21], and GASTON [19]. Asbru also allows the formal specification of the guideline's intentions regarding process and outcome. This allows assessment of compliance of the physician's actions and the patients' outcome not only to the exact guideline recommendations but also to the intentions behind them; it is thus possible to evaluate whether the actual treatment was in accordance with the guideline intentions even if it deviated from actual recommendations.

Beyond the representation of clinical algorithm logic, factors that are important for successful CIG implementation include integration of CIGs with (a) institutional information systems and (b) institutional workflow, and (c) support for local adaptation of generic guideline recommendations. The first two topics are covered in Section 4.3 (Integration of CIGs with EHRs and organizational workflow). Workflow integration is specifically addressed by the GUIDE [27–29] and SAGE [31] CIG formalisms. Stressing the importance of fitting CIGs into organizational workflow, Quaglini, Stefanelli, and their colleagues coined the term careflow [27]. In [27], the authors describe their use of the New-

Guide formalism for developing an evidence-based careflow management system for post-stroke rehabilitation in a hospital in Italy. The system involves different organizational units, such as wards, rehabilitation units, clinical laboratories, and imaging services. Several organizational agents work within these units and play one or more roles. Each role is defined by the set of goals that she/he must fulfill. The knowledge-base includes knowledge for managing exceptions, which may occur in daily medical work as any deviation from the normal flow of activities. It allows either avoiding or recovering automatically from expected exceptions. Recognizing the importance of flexibility [28], organizational agents are allowed to modify the scheduled flow of activities for an individual patient under the only constraint of justifying their decision.

The workflow integration approach of SAGE is based on identifying scenarios (context) for which decision-support could be provided. The context is characterized not only by a patient state but also by a triggering event (e.g., abnormal test result was obtained for a patient), organizational setting and the involved organizational roles.

The third important factor for successful implementation is support of local adaptation of CIGs. This is needed when a guideline is to be implemented in a different institution. The adaptation is needed because of different organizational setting, policies and

workflows, and sometimes even because of differences in patient populations or pathogens and environmental conditions specific for the local setting. The CIG formalism should contain mechanisms for adapting guidelines and for tracking the changes among guideline versions (see also Section 4.7 devoted to maintenance). All of the TNM formalisms use a hierarchical complexity management mechanism (called "nesting") to specify the details of tasks belonging to higher-level clinical algorithms in separate lower-level sub-guideline instances. As suggested in [23], different sub-guidelines could be developed for each local institution adapting the guidelines. In this way, higher-level views of the guideline algorithms can be shared while allowing specific details of workflow to be different at lower-level algorithms (sub-guidelines).

One of the important goals of the developers of GLIF3 [25] was to develop sharable CIGs where each CIG could be adapted to different local implementation environments. They have deliberately studied other CIG formalism and incorporated into GLIF3 support for those functionalities that they perceived as most useful for diverse guideline implementations as well as extending it to annotate clinical concept with standard vocabulary terms, patient information models and expression languages to aid in the task of linking CIG terms to EHR data [24].

Standards play an important role in the ability to share CIGs as they encourage vendors to develop commercial applications for CIG-based DSS. The object-oriented patient information models of GLIF3 and SAGE are derived from the standard Reference Information Model (RIM) of HL7. It includes a small subset of classes consisting of Substance Administration, Observation, and Procedure that could be applied to a patient. For a related paper discussing an earlier version of the RIM and its alignment with clinical guidelines and with EHR data please see [32].

Drawing lessons learned from the Arden Syntax, GLIF3 uses two alternative expression languages to formulate decision criteria. The earlier expression language is Guideline Expression Language, GEL [24], which is derived from the Arden Syntax and uses its logical and temporal operators, while providing extensions. The later expression language is Guideline Expression Language Object-Oriented (GELLO) [33]. This language is used to specify decision criteria, formulae, and constraints on data and action and is an HL7 standard since 2004. Unlike GEL, GELLO's object-oriented model fits well with the object-oriented HL7 RIM. This makes it easier to specify guideline decision criteria and expressions over patient data that are structured according to GLIF3's RIM-based patient information model. GELLO is based on the Object Constraint Language (OCL) (www.omg.org/spec/OCL/2.0/), is vendor-independent, object-oriented, and side-effect-free. While GELLO can work with any object-oriented data model, standardization is facilitated by ensuring that the data model is compatible with the HL7 RIM.

4.2. Knowledge acquisition and specification methods for improved quality of CIG design

While all CIG languages aim to support clear, complete, and correct structuring of a narrative CPG as a CIG that would be the basis of a DSS, many researchers have come to the conclusion that the introduction of CIG languages is a necessary yet not sufficient instrument needed to yield high-quality CIGs. Thus, they have suggested different methods and tools to support the CIG design process, increasing the quality of this process and of the generated CIGs.

4.2.1. Using cognitive methods to support translation of CPGs into CIGs

Patel and colleagues [34] explain how methods of cognitive analysis could support the design and evaluation of CPGs. Cognitive research methods examine how people mentally represent information and subsequently use it for problem solving. The problem-solving task studied in this work is the translation of CPGs into CIGs. The authors start by describing the difficulty in converting a narrative CPG into a formal CIG: "CPGs can be semantically complex, often composed of elaborate collections of prescribed procedures with logical gaps or contradictions that can promote ambiguity and hence frustration on the part of those who attempt to use them". Their proposed solution is to use cognitive analysis in order to improve the understanding of the semantics and structure of CPGs and correct it where necessary. The resulting CIGs would thus be free of contradiction, ambiguity, correct, and more complete.

Cognitive analysis takes as input narrative guidelines and identifies and formally encodes the propositions that are expressed. Such (manual) analysis can identify ambiguous areas in the text that lead to misunderstanding. The analysis can uncover different interpretations of ambiguous text, which go against the very aim of CPGs: to reduce unjustified practice variation. The variability in interpreting CPGs can be decreased by converting complex propositions to sets of single propositions that are logically linked together. Implicit knowledge, known to clinical experts but not to knowledge engineers, can be optionally added to provide context and narrow inference gaps. After employing such a process, the narrative text could correctly be converted into formal CIG representation. Once a CIG is specified, cognitive analysis methods could be used to assess CIG adequacy as reflected in the thoughts, confusions, or solutions achieved by users.

Peleg et al. [35] examined the process by which a medical expert from the American College of Physicians (ACP) created paper-based clinical algorithms from CPGs. The authors studied this process by looking at intermediate versions produced during the algorithm creation. They identified and analyzed errors that were generated at each stage, categorized them using Knuth's classification scheme of programming code errors, and studied patterns of errors. Possible explanations for the sources of these errors were assessed and recommendations for reducing errors were provided, based on cognitive theory and on experience drawn from software engineering methodologies. The authors' concluded that team-work is needed in order to identify errors in CPGs as well as cognitive errors introduced during algorithm creation such as overlearning and confusing different situations. Including an informatician on the team can help in identifying computability errors, such as missing definitions of branching points and interaction among guidelines, problem with negation and implication, and confusing AND with OR. The authors further suggested that authoring tools for the GLIF3 [25] CIG formalism could be employed during algorithm creation to promote computability.

A related study [36] came to similar conclusions. The authors analyzed and categorized the errors found during the process of translating a paper-based guideline into an electronic format and in a second step they used a process-modeling tool to revise the content of the guideline. Their conclusion was that guideline development should be assisted by process modeling tools, which check the content in comparison to the process meta-model. The meta-model itself could support the domain experts in formulating their knowledge systematically.

Two other studies by Patel and colleagues used cognitive methodologies to examine the process of translating narrative CPGs into CIGs. The earlier study [37] examined how individual and collaborative expertise affect representation of CPGs in the GLIF2 [38] CIG language (an earlier version of GLIF3). A research question was whether clinicians working alone or knowledge engineers working alone could successfully translate a CPG into a CIG and what would be gained from their collaboration.

The cognitive research methods used by Patel involve detailed analysis of a few subjects, and generalization of cognitive steps underlying observable behavior, focusing on individuals' specific organization of knowledge structures and sequences of reasoning steps. First, a cognitive task analysis is carried out by specifying the knowledge and information processing capabilities humans have and use in a specific task. For this study [37], the aim was to learn how individuals with different background and skills (e.g., clinicians and knowledge engineers) create internal representation of the CPG knowledge that are intermediate representations before creating the final flowchart representation of the clinical algorithm. Second, a detailed, predictive model is developed. In this study [37], the prediction was that the internal representation of the CPG will contain more details than the CPGs itself, will reorganize the knowledge, and will resolve ambiguities. Given the different knowledge and skills of clinicians and knowledge engineering, it was also predicted that their internal representations would be different. Finally, clinicians and knowledge engineers were observed and were asked to describe their thoughts while they read the CPG and created the GLIF2 model. This think-aloud is assumed to reflect information that individuals have in working memory during problem solving. The researchers analyzed the participants' verbalized interpretation of the CPG as well as their final GLIF2 representations to infer the content and structure of the participants' internal representations. The final GLIF2 representations were compared to results from the task analysis. The extent to which the observed data fit the theoretical model is taken as evidence of the psychological validity of the model.

Patel's study showed that the representations developed by physicians used implicit medical knowledge and a process flow structure not explicitly stated in the guidelines, reflecting their understanding of the underlying pathophysiology. The knowledge engineers developed more literal representations of the guideline, making additions that were mandated by the logic of language. Their collaboration yielded clearer, more complete, and more correct representations, where both domain-specific knowledge and generic knowledge of guideline structures were integrated.

In a later study [39], Patel and colleagues analyzed the process of encoding two clinical guidelines into GLIF3 [25] and into GLIF2 by two medical informaticians (knowledge engineers). GLIF3 is a later version of GLIF2 that supports specification of computable decision criteria and linking of a CIG to patient data from an EMR and was hypothesized to be more expressive and robust in specifying the content and logical structure of CPGs. The researchers recorded a video of the two modelers as they were using authoring tools to create the GLIF2 and GLIF3 specifications. The video was created by remotely logging into the authoring application and recording the screens displayed while also recording the verbal explanations provided by the modelers as they were working with the authoring tools. The videotapes of the modelers' interactions as they encoded the guidelines were transcribed and coded based on their encoding-related actions and verbalizations. The coding scheme identified and classified the modelers' encoding-related actions, such as the constructs used to represent the guideline in GLIF as well as problems that were experienced during the encoding. The formative evaluation demonstrated that GLIF3's intended improvements in expressiveness were achieved.

4.2.2. Modeling methodologies and tools for translation of CPGs into CIGs

The results of Patel's studies have made major implications for development of CIG modeling methodologies. Several research groups have suggested various ways for conducting the collaboration between clinical experts and knowledge engineers. For example, Shalom, Shahar, and colleagues [40] suggest and report

an evaluation of a collaborative specification and evaluation of knowledge acquisition methodology for clinical guidelines, which is referred to as the "consensus method". This methodology, which was evaluated in three healthcare settings, including the VA Palo Alto Health Care System, is used to create CIGs in the DeGeL guideline library discussed in Section 4.5.2. It includes nine steps, performed by a team of expert physicians (EP), clinical editors (CE), and knowledge engineers (KE). The methodology contains seven steps for translating a CPG into a CIG and two additional steps used for evaluation. In the most crucial step in this methodology, which has been shown to make a critical improvement of the resulting CIG specification, the team creates a high-level agreed-upon consensus-based structuring of the guideline by focusing first on a clinical consensus about the CPG semantics and then on an ontology-based consensus according to the CIG language selected for knowledge representation. The evaluation of the consensus methodology has shown that given an ontology-specific consensus, clinical editors with mark-up training can structure CIG knowledge with high completeness, yet correct structuring requires thorough training in the ontology's semantics. The gradual-specification process supports different types of users: EPs, medically trained CE who mark-up the GL, and knowledge engineers, typically informatics experts who can create a formal GL representation in a later step. The incremental, hybrid knowledge-specification methodology is supported by the Gesher tool, which is implemented as part of the DeGeL guideline library reviewed in Section 4.5.2.

To assist in creating CPGs that are clearer and more implementable as CIGs, Shiffman and colleagues developed Building Recommendations In a Developer's Guideline Editor Wizard (BRIDGE-Wiz) [41] – a software assistant that promotes clarity, transparency, and implementability of narrative CPGs required to create CIGs and evaluated its use with guideline developers who used it in five guideline-development efforts and found it to be useful and usable. BRIDGE-Wiz provides a template to create structured natural language recommendation statements addressing the questions: (1) under what circumstances? (2) who? (3) ought (with what level of obligation?) (4) to do what? (5) to whom? (6) how and why? BRIDGE-Wiz helps clarifying recommendations by limiting the choice of action types and of verb types to a set that was identified by studying more than 700 recommendations from the Yale guideline recommendation corpus, building active voice recommendations, and limiting Boolean connectors.

To enhance transparency, BRIDGE-Wiz users are directed to check the executability of the action and decidability of the conditions under which it is to be performed based on the GuideLine Implementability Appraisal (GLIA) instrument [42]. GLIA contains quality assessment questions arranged into categories that can be applied to guideline recommendations in order to appraise evidence quality, benefits, and harms, and identify implementation barriers. BRIDGE-Wiz promotes implementability by providing a pseudocode rule, suggesting deontic modals ('must', 'should', 'may') that define the intended level of obligation, and limiting the use of 'consider'.

BRIDGE-Wiz builds upon the methodology [43] proposed by Shiffman and colleagues to systematize and make explicit the translation of document-based knowledge into workflow-integrated clinical decision support systems. This methodology turns a markup of selected CPG text into coherent and complete recommendations by de-abstracting and disambiguating recommendation concepts, verifying rule set completeness, adding explanations, and building executable statements. This methodology uses the Guideline Elements Model (GEM) [7] reviewed in Section 4.1. GEM is supported by many tools, including GEM-Cutter (http://gem.med.yale.edu/), for marking up guidelines according to GEM elements.

Additional Markup tools were developed, which are not tied to a particular CIG formalism, which support step-by-step transformation of medical guideline documents into a formal (or even operational) representation. They include the Stepper tool [44] and the Document Exploration and Linking Tool (DELT/A) (http://ieg.ifs.tuwien.ac.at/projects/delta/).

Another approach for structuring CIGs [45] proposes the use of clinical design patterns in order to conceptualize and standardize the representation of particular classes of clinical guidelines as CIGs, using domain-level abstractions. The authors have demonstrated their approach by developing design patterns for screening guidelines and for immunization guidelines. Screening guidelines patterns include visual templates that structure screening guidelines as algorithms of guideline steps used for screening and data collection. Immunization templates focus on representation of scheduling constraints for the different immunization doses.

4.2.3. Information extraction methodologies and tools for translation CPGs into CIGs

Information extraction methods have also been developed to support semi-automatic translation of narrative to marked-up text. Serban and colleagues [6] defined linguistic patterns, demonstrated how they could be instantiated in the text of a guideline for treatment of breast cancer, and evaluated the usefulness of these patterns in the modeling of this guideline. In [46] knowledge templates were combined with medical background knowledge from existing thesauri in order to produce reusable building blocks used in guideline development.

Linguistic patterns are also used by the LASSIE tool [47] to extract from CPGs clinical actions that can be marked up using the DELT/A tool. Moreover, the information extraction methods used rely on patterns [48] at the level of a phrase (i.e., regular expression that build the attributes of actions from basic entities, such as time, dosage, iteration, and condition expressions), a sentence (i.e., phrase patterns extended with medical terms, and trigger words for the medical terms to identify medical actions and their attributes), and discourse (i.e., sentences are categorized, merged to actions, and relationships between actions are established). The intermediate representation of guideline knowledge at the level of extracted patterns is independent of the final formal representation in some CIG formalism.

4.2.4. CIG authoring tools

For CIG formats that have XML-based syntax (e.g., XML, RDF/XML) generic XML editors or ontology editors such as Protégé (protégé.stanford.edu) can be used to author them, as is the case for EON, SAGE, and GLIF3. The Protégé editor and most other custom-made editors, visualize a clinical algorithm as a network of tasks or flowchart. In contrast, AsbruView [49] uses a unique user interfaces consisting of metaphors of movement to visualize Asbru CIGs, stressing the time-oriented and skeletal-plan refinement nature of Asbru CIGs. Graphical metaphors that include running tracks and road and track signs are used to communicate temporal information and topological relationships between plans.

4.2.5. Can CPGs and CIGs be developed concurrently?

The studies mentioned above have highlighted the sources of ambiguity, lack of clarity and incompleteness of CPGs that introduces barriers into their translation into CIGs. Most of the methodologies described above have used existing CPGs as the starting point of developing CIGs and employed teams of clinical experts and informaticians for disambiguation and creation of correct and more complete CIGs. However, it seems logical that use of tools and introduction of informatics computability considerations into the CPG creation process could result in the creation of narra-

tive guideline recommendation that are at the same time directly computable. Two studies have put this hypothesis to the test. In the BRIDGE-Wiz study [41], Shiffman has demonstrated that structured recommendations could be authored using the BRIDGE-Wiz tool and manually inspected to assess the potential for executability and implementability.

The study by Goud et al. [50] went even further by developing a CPG for cardiac rehabilitation and its concurrent formalization in the GASTON [20] CIG formalism. The authors have synthesized a guideline-development strategy based on an analysis of existing methodologies for CPG development and CIG formalization. In their strategy, guideline formalization is performed concurrently with guideline development. The strategy involves guideline formalization specialists and formalization tools in early stages of the development process, cooperation between guideline authors and guideline formalization specialists in the development of clinical algorithms, access to domain knowledge when formalization identifies inconsistencies or omissions, and formal verification of the guideline model prior to guideline dissemination.

4.3. Integration of CIGs with EHRs and organizational workflow

Studies have shown that clinical decision-support systems should be integrated into organizational workflow and should be integrated with the organization's information systems, and in particular with EHRs, in order to make their use easy and efficient [51]. CIGs are not always integrated with EHRs and organizational workflow although this generally improves their effectiveness. In addition, consideration of organizational resources and scheduling constraints (organizational workflow) are important for supporting not just clinical decision-making but also action management. In this section I address these two issues.

4.3.1. Integrating CIGs with EHR data

CIGs provide patient-specific advice by matching CIG concepts (contained in decision criteria) with patient data. Patient data can be supplied by the user. However, data entry is tedious and error prone. Hence, if up-to-date data is contained in electronic medical records, allowing guideline engines to match that data to CIG decision criteria is beneficial.

Different approaches to the knowledge-data mapping have been proposed. The knowledge-data ontological mapper (KDOM) [52] closes semantic gaps between the CIG knowledge and EHR data by defining abstractions from the more specific data (usually contained in EHRs) and more general concepts (usually found in CIGs). Abstractions are defined by combining different types of mapping classes, including simple temporal mapping, hierarchical mapping, and logical combination mapping. CIG knowledge and EHR data are both mapped to a global schema view that conforms to a standard patient data model, following the global-as-view approach to data integration, where the common data model is used to generate standard views of proprietary EHRs. KDOM uses declarative query mapping supported by a meta-language for defining mappings (a mapping ontology) and an SQL Generator that translates mapping instances into SQL queries used to retrieve the corresponding patient data. KDOM has been evaluated by mapping a GLIF3-encoded guideline into two different EMR schemas and by using the mapping ontology to define mappings from 15 GLIF3 CIGs and one SAGE CIG into an HL7-based RIM model.

The standard patient data model used in KDOM is based on HL7-RIM-based models, such as the Virtual Medical record (vMR) model. The vMR is an HL7-RIM based standard based on the seminal work proposed in [53], especially designed for the purpose of

integrating patient data with DSSs. It includes a small number of classes derived from the Clinical Statement class that represent clinical problems, observations, adverse events, goals, encounters, procedures, substance administration, and supplies. The small number of vMR classes provides an easy-to-use conceptual model of EHR data used for decision-support.

In the KDOM paper described above, locating the EHR data fields that match CIG concepts was done manually. Terminology services, such as Medical Database Adaptor (MEIDA) [54] can assist in this task.

4.3.2. Workflow integration to support action management

Organizational considerations and workflow adaptation are crucial for the success of CIG-based DSSs. Several CIG formalisms address these issues. One of them is NewGuide [27–29] in which decision-support is achieved by linking together the guideline management system, the EHR, and the workflow management system (WfMS). The WfMS manages the health care activities according to organizational knowledge related to organizational roles, resources, and schedules (e.g. nurses' and physicians' shifts scheduling). Workflow management includes maintaining the correct CPG flow and timing as well as retrieval of missing data from the EHR is left to external WfMS.

Modeling guidelines in SAGE [31] includes identification of usage scenarios (contexts) of guideline-based care in clinical workflow and encoding them and their appropriate guideline recommendations. Contexts are characterized by a triggering event (e.g., patient checking in), patient characteristics, organizational setting (e.g., primary care outpatient clinic), organizational roles specifying who should respond to the event (e.g., nurse), and needed resources—coordinate the activation of guideline-based decision support.

The workflow integration approach used in [55] translates an Asbru CIG into temporal hierarchical task networks (HTNs). Through planning and scheduling, the HTN model is used to create patient-specific clinical pathways that take into account complex temporal constraints and the allocation of institutional resources. The method focuses on the representation and management of temporal knowledge and is based on the identification of workflow and temporal patterns in a CIG.

4.4. CIG validation and verification

Because CPGs and their CIG implementations are meant to decrease errors and increase quality and safety, an important step in their life-cycle includes validation and verification. Surprisingly, the terms validation and verification are not clearly differentiated with respect to CIGs. I use here the definitions provided by Boehm [56] where validation answers the question "are we building the right product?" and verification answers the question "are we building the product right?". In the rest of this section I review research concerning validation and verification of CIGs, as well as the use of formal verification methods for assessing expressivity of CIG languages.

4.4.1. CIG validation

In the case of CIGs, the clinical experts serve as the customers whose requirements are captured in CPGs. Here, validation concerns a process whose purpose is to establish that the CPG's requirements are captured in the CIG specification.

Two main validation techniques can be used by clinical experts assisted by knowledge engineers. In the inspection technique, clinical experts inspect the CIG to try and locate errors in the clinical algorithm's logic. Additionally, when different properties and quality indicators are formally represented for the purpose of verification (discussed in Section 4.4.2) the experts should also inspect and validate that these properties are relevant and are correctly defined, and that all of their desired properties have been formalized. However, medical experts find it hard to exhaustively check the CIG's formal decision criteria that are often quite complex in terms of logical combinations of individual patient data items. Hence, errors in decision criteria and in procedural knowledge or temporal knowledge are often still found after such validation by inspection took place. Therefore, validation by experts [40,57] also includes a second technique – testing – in which experts (who could be assisted by knowledge engineers) test the CIG with different (simulated or real) patient data values to determine whether the recommendations output by the CIG execution engine are in line with the CPG's recommendations.

In this mode of validation, the clinical experts think about the actions that they would have recommended for the patient case. They do not need to procedurally follow the steps of the algorithm but evaluate recommendations generated by the CIG execution engine at key points of the guideline. Hence, testing serves as validation of the correctness of the CIG's output against medical knowledge (e.g., CPG knowledge); it does not check the CIG's specification. To increase validity, validation is done by several clinicians, including those who were not intensively involved in the team that created the CIG. Using cognitive methods, experts are sometimes asked to "think aloud" (i.e., to verbalize their thoughts) while thinking about the patient case and while working with the CIG execution engine on particular patient cases. This methodology allows to gain insights into experts' cognitive processes and to identify potential cognitive pitfalls in the CIG.

In order to be confident that a CIG was thoroughly tested, different branches of the clinical algorithm should be traversed and executions with missing values should also be checked. As argued by Miller [58], validation through testing needs to be applied each time the guideline is updated. For these reasons, automatic generation of test cases can support rigorous, scalable testing. Miller presents a case study of the IMM/Serve – a DSS that is based on childhood immunization guidelines. The national CPGs from which this DSS has been created, has evolved over the years and it was therefore necessary to maintain the CIG knowledge-base up to date. As medical knowledge evolved and as users requested changes in the CIGs, the developers of this system developed a method for automated generation of test cases that can be used for revalidation.

Although the representation of CIG knowledge in the IMM/Serve DSS is done using tables of temporal parameters, if–then–else rules and procedural logic, the testing techniques used by Miller could also be applied to CIGs that are based on TNMs, as the three components used for knowledge representation are also found in TNM representations.

To test the logic of the CIG representation, Miller uses several techniques. One technique uses tools to test the knowledge base directly to find potential inconsistency and redundancy. The second technique is to generate test cases based on the clinical algorithm's logic by considering patient scenarios: combinations of values of patient data items that the CIG's expressions refer to. This technique is also used in [59]. In IMM/Serve, users of the T/Gen tool [60] can specify domain-specific constraints as to which combinations of patient data items do not make sense or do not need to be exhaustively tested against one another. The third technique is to use formalized domain knowledge external to the CIG logic to generate test cases. This external knowledge usually includes (1) knowledge about practice customs, preferences and expectations and (2) temporal knowledge

about the healthcare process that is not part of the guideline. In the case of IMM/Serve, the external knowledge used was specific to the immunization process. However, this idea could be generalized to the idea of Basic Medical Knowledge proposed by Terneziani et al. [61] (i.e., general medical knowledge implicit in the CPGs – assumed to be known by physicians using this guideline – that can be used to reason with explicit CPG knowledge).

4.4.2. Formal verification of CIGs

Unlike validation, which concerns informal evaluation by domain experts, formal verification concerns mathematical proof that an implementation meets its formal specification. In other software development processes, the design specification is often distinct from its implementation code, hence it can be verified that the implementation meets the design specification. However, in development of CIG-based DSS, the CIGs serve as both the formal specification and as the implementation, provided that their execution engines use the correct semantics of the CIG language. Therefore, verification of CIGs falls into two categories: (1) proving that the CIG specification is internally consistent and free of anomalies, and (2) proving that the CIG specification satisfies a set of desired properties, which are formally defined. These two verification categories refer to verification of single CIGs. A third category of verification concerns (3) checking inconsistencies between CIGs that are concurrently applied to a patient with comorbidities. In this subsection I also address verification techniques for assessing expressivity of CIG languages.

4.4.2.1. Verification: proving consistency.
Knowledge-based flaw (anomaly) detection is an approach for proving consistency that has been developed by Duftschmid and Miksch [62]. In this approach, guideline-specific knowledge needed in order to detect flaws in Asbru CIG specifications is specified in a knowledge base. The verification is not based on execution of Asbru CIGs but on static examination of the plan conditions specified in CIGs against the anomaly detection knowledge. Their methods allow the detection of anomalies within single components (level 1), single plans (level 2) and whole plan hierarchies (level 3). By verifying plan conditions, the following anomalies can be detected: level 1 anomalies, including unsatisfiable conditions and redundant parameter-value pairs within conditions, level 2 anomalies, including unreachable valid sequence of plan states and ambiguous state transition, and level 3 anomalies, including inability to complete and termination enforced by parent.

In later work, Duftschmid et al. [63] focused on the detection of flaws within temporal scheduling constraints of Asbru CIGs. Their verification method is based on calculating the minimal network of temporal constraints on the execution of guideline activities. The method checks whether temporal scheduling constraints are consistent with scheduling constraints implied by control flow operators and the hierarchical structuring of a guideline. It suggests equivalent, yet more explicit representation of non-minimal constraints. Furthermore, it can be used by the guideline interpreter to assemble feasible time intervals for the execution of each guideline activity.

4.4.2.2. Verification: satisfying properties.
Checking that desired temporal-logic properties hold for a CIG specification (i.e., for the potential execution pathways that the CIG enables) can be done using a variety of techniques, including theorem proving and model checking. In both of these methods the CIG is first translated into another formal specific method that can be used by a model checker or theorem prover.

4.4.2.2.1. Model Checking. Model checking involves exhaustive checking of all pathways of an executable specification. Pérez and Porres used a model-driven approach for authoring and verification of clinical guidelines [64]. They do not use one of the formal CIG languages but instead use Unified Modeling Language (UML) Statecharts to represent the dynamics of guidelines, and compile the guideline representation to generate a clinical guideline-based decision support system [65]. Their Statecharts-based DSS can provide guided support to the physician during guideline application and automatically stores guideline application data for traceability purposes.

To support verification, the authors use a Model Driven Development (MDD)-based tool to automatically process the Statecharts representation to generate the input model of a model checker. The model checker verifies the resulting final state transition model against specific required temporal-logic properties that should hold for the CIGs and if the properties do not hold it generates counterexamples.

The authors have established a pattern-based hierarchy for defining commonly occurring types of requirements in guidelines, drawn from a thorough literature survey in different domains (not limited just to the medical domain). The pattern hierarchy includes occurrence patterns related to CIG paths. Occurrence patterns relate to the existence or absence of certain states/events during a defined interval of time. States may represent goals of the clinical guideline that should be met and events may represent clinical actions that should be followed [66]. The occurrence requirements originate from (1) good medical practice (e.g., precluding the prescription of redundant drugs, or advising against the prescription of a treatment that is less effective than some alternative), (2) particularities of implementing institution (e.g., make sure that in the case of a specific set of resources being available (or not available) there is a therapy for a patient to which a guideline is being applied), (3) CPG goals (e.g., establishing whether paths lead to the CPG's goals), and (4) patient clinical state (e.g., proving whether there is a therapy for a patient with a specific clinical state). The authors demonstrated their approach in the context of a guideline for the management of catheter related infections.

One of the limitations of the work by Pérez & Porres is that they have not used a special CIG language but relied on UML Statecharts which do not contain an expressive language for temporal expressions which often occur in CPGs. On the contrary, the research group of Tereneziani et al. [67] proposed a way to verify an ischemic stroke guideline represented as a GLARE CIG using the SPIN model-checker that supports checking of temporal constraints expressed in Linear time Temporal Logic (LTL) formulas. Structural and medical validity, contextualization, applicability, and complex properties like paths of actions or sequences of patient's states were checked with this approach.

4.4.2.2.2. Theorem proving. Theorem proving explores logical derivations of a theory. ten Teije and co-authors [68] have used a theorem prover to confirm whether a protocol for management of jaundice in newborn babies that was specified in the Asbru CIG language and then translated into a formal KIV representation complies with certain protocol properties. KIV is an interactive theorem prover with strong proof support for higher-order logic and elaborate heuristics for automation, appropriate for verifying large protocols. Using this approach, they verified properties such as plan termination, temporal action intentions and indicators published by the Making Advances against Jaundice in Infant Care Committee. When the authors tried to prove that treatment plans comply with one of the published indicators, they uncovered a problem in the Asbru protocol's formalization,

demonstrating the usefulness of verification for establishing CIG correctness.

4.4.2.3. Verification of concurrent CIGs for patients with comorbidities. Wilk et al. [69] propose a method that verifies if two CPGs can be applied concurrently to a patient with comorbid diseases and that uses conflict resolution to arrive at a possible revision. The method was illustrated in the context of two clinical scenarios involving a patient with duodenal ulcer who experiences an episode of transient ischemic attack. The mitigation algorithm starts by representing guidelines as logical models. Then the models are solved to formulate a feasible combined therapy. If there is no feasible solution, then interaction operators, which characterize adverse interactions, are used to identify the source of infeasibility. Next, revision operators are used to describe possible revisions to the logical models. The interaction and revision operators are based on medical knowledge that is external to the CPGs, and may be thought of as basic medical knowledge [61] (see Section 4.4.1 for a definition of basis medical knowledge). Aiding in efficiency management, direct adverse interactions are handled by the algorithm before it addresses indirect interactions and only after it can mitigate direct interactions.

Despite the importance of the topic of integrating CIGs for patients with comorbidities, the search queries in this review did not retrieve additional papers addressing this challenge. This topic is revisited in the Discussion section (Section 5.1).

4.4.3. Verification for assessing expressivity of CIG languages

Another formal approach has been suggested by Grando et al. [70], focusing on the analysis of the control-flow expressivity of CIG modeling languages. A previous study compared CIG languages according to their ability to support workflow (control) patterns without formal proof [71]. To prove that a CIG language satisfies or does not satisfy a workflow pattern, the pattern is first represented as a colored Petri Net, *P*. To prove that a CIG language can support the pattern, the CIG language is used to specify the pattern, *W*, which is translated into a colored Petri Net. The translation can be automatic if the language has been mapped to a Petri Net model. To prove that *P* and *W* have similar behavior, a Petri Net tool is then used to calculate the state-space graphs of *P* and *W*. Edinburgh Concurrency Workbench tool is applied over the state space graphs to automatically determine if they satisfy any of the bisimilarities specified in process calculus by Milner. To prove that a CIG language does not satisfy a pattern, all the different ways in which a pattern could potentially be represented in the CIG language are explored and if no bisimilarity is satisfied then the proof by exhaustion is valid.

4.5. CIG tools and execution engines

In order to be operational, any CIG language should have at least two tools: a CIG authoring tool (knowledge acquisition tool) and a CIG execution engine. While CIGs can be edited using generic editors (Section 4.2.4), to be executed, special-purpose CIG execution engines were developed for the different CIG formalisms [72]. In this section, we review CIG execution engines and additional CIG-related tools, including search tools and CIG library tools.

4.5.1. CIG execution engines

The GLIF3 Guideline Execution Engine (GLEE) [73] provides defined interfaces to electronic medical records (EMRs) and other clinical applications to facilitate its integration with the clinical information system at a local institution. The execution model makes recommendations but the user is the one making the final selection of decision alternatives. GLEE supports both interactive synchronous mode and an event-driven execution model, once it is linked to the clinical event monitor in a local environment. It includes a tracing system to record an individual patient's state when a guideline is applied to that patient and allows multi-encounter decision-support that can continue from the last saved state but can also start from any entry state into the CIG. GLEE's API has been used to implement a GLIF3 CIG-based DSS based on a clinical guideline for management of diabetic foot infections [57]. The DSS was linked to an electronic medical record developed by RAMBAM Medical Center in Israel.

GLIF3 guidelines can also be executed by Guideline Execution by Semantic Decomposition of Representation (GESDOR) [74] – a generic execution model for sharing of computer-interpretable CPGs. GESDOR's generic execution model supports generic execution tasks: (1) primary tasks, such as data collection, clinical intervention, medical decision making, patient state verification, branching, synchronization, and subguideline, and (2) auxiliary tasks, such as criterion evaluation, event registration, and event invocation, which are used to support the execution of the primary tasks. By mapping different CIG languages to these generic tasks, execution of different CIG languages is enabled, thereby supporting sharing at the executable level. GESDOR has been used to execute CIGs specified in GLIF3 and PROforma.

The Spock execution engine [75] is an engine designed as a client–server architecture that can execute guidelines represented in Hybrid-Asbru intermediate representation (see Section 4.2.2). Spock can execute the representation against electronic patient record or ask the user for the needed input. The Spock system was integrated into the Digital electronic Guideline Library (DeGeL) framework, reviewed in Section 4.5.2. For evaluation, three guidelines were specified in an intermediate format and were applied to a set of simulated patient records designed to cover prototypical cases that were correctly executed.

Other approaches to CIG execution, such as that of GASTON, includes application of problem-solving methods. The GASTON toolset [76,77] includes a knowledge acquisition tool for authoring domain ontologies and problem-solving methods (PSMs – see examples in Table 3) and an execution engine. The knowledge-acquisition tool takes the control structure of each guideline and PSM and creates a structure consisting of primitives. This structure is then combined with the implementation procedures that are attached to each primitive, and is automatically compiled into a more efficient description that can be executed by the execution engine.

4.5.2. Other CIG-related tools

Two CIG supporting tools from Shahar's research group include the Digital Electronic Guideline Library (DeGeL) [78] and the Vaidurya clinical-guideline search engine [79].

Digital Electronic Guideline Library (DeGeL) is a framework for a distributed, hybrid, multiple-ontology clinical-guideline library, and automated guideline-support tools [78]. DeGeL has a web-based, modular, distributed architecture, which facilitates gradual conversion of clinical guidelines from text to a formal representation in a selected target guideline ontology. The architecture supports guideline classification, semantic markup, context-sensitive search, browsing, run-time application, and retrospective quality assessment. The feasibility of the architecture and the tools has been demonstrated for the Asbru and GEM CIG formalisms.

Addition of new guidelines into the guideline library requires the classification of the guideline along several semantic hierarchical metadata axes. To ease this task, an automatic classifier has been developed, which classifies new guidelines along these axes,

based on a given set of guidelines previously classified using the same axes [80].

Once a large number of guidelines are in the library, it becomes non-trivial to locate a guideline that fits the user's needs. To support such retrieval, a search engine called Vaidurya [79] has been developed as part of DeGeL. In addition to full text search, two search methods were implemented: (1) concept-based search, which relies on pre-indexing the guidelines according to the semantic axes of DeGeL, and (2) context-sensitive search, which relies on first semi-structuring the guidelines according to a given ontology (e.g., Asbru), then searching for terms within specific labeled text segments. Evaluation of Vaidurya demonstrated that both search methods are feasible and significantly improve on free text search retrieval performance.

4.6. Exception and error handling

The appropriate management of exceptions is an important topic to address as they may arise during CIG execution and in some cases may cause medical errors. In those cases, the DSS should be able to detect the exception and respond accordingly. Grando and colleagues have developed a goal-oriented framework for specifying clinical guidelines and handling medical errors [81]. This framework uses a state-based model to link plans, goals, monitored effects, and exceptions by extending the PROforma CIG formalism with Petri Net semantics for scheduling constraints. It enables specifying the goals of a guideline and linking them with recommended tasks that could satisfy the goals. Exceptions are linked with goals that manage them, which can be realized by tasks or plans and goal-based planning is used to suggest tasks for handling exceptions. Exceptions are specialized into obstacles and hazards. Hazards are potentially harmful states that occur at run-time and may be a threat to patient safety. Obstacles on the other hand prevent guidelines from being followed or from achieving their intended effect. The approach was demonstrated using a generic plan for management of a chronic disease and a particular instantiation for hypertension management.

Two subsequent papers by Grando and colleagues extended the previous work. In [82], goal-based patterns for collaborative work, which is very common in healthcare, were defined. The patterns are intended to provide generic and reusable solutions that are flexible enough to be customized at run time to detect exceptions in team work and to transfer responsibility and accountability for recovering from exceptions to the appropriate actor. Using Owicki–Gries theory, the authors have proved that the proposed patterns satisfy the properties that characterize service assignment and delegation in terms of competence, responsibility and accountability in normal and abnormal (exceptional) scenarios. The proposed patterns were also instantiated in an executable COGENT prototype and they can be mapped into the Tallis tool that enacts PROforma language specifications of clinical guidelines.

In a later paper, Grando and colleagues [83] considered more expressive specification of actors, roles and actor constraints, by adding the notions of patient and healthcare organizations, and by adding an argumentation-based decision support system for advising clinicians at decision points concerning choices on what to believe and what to do. They have fully implemented the framework in the COGENT formal modeling system and have evaluated it by implementing a hypertension guideline. The implementation demonstrated adaptation to a health organization's resources and a patient's particular medical condition, delegation of health care, and re-planning when unexpected situations arise.

4.7. CIG maintenance: compliance and versioning

The maintenance phase begins after the CIG is already in use. During this phase, the institutions that implemented the CIGs monitor its usage and impact, collect requirements for necessary changes to the CIG's logic, and try to correct and improve the CIG and keep it up to date with new versions of CPGs that are being created [58].

CPGs aim to base medical care on evidence, reduce unjustified practice variation, increase healthcare quality, and reduce costs. However, evaluating whether their CIG implementations can yield such impact is complicated by the fact that CIG implementations provide flexible decision-support, allowing clinicians to deviate from CIG recommendations, and has been done in very few cases.[1] Therefore, a necessary and more feasible step is to demonstrate that CIGs impact clinicians' behavior, improving their compliance with clinical guidelines recommendations. Other evaluation studies with users that assess the usability of the CIG-based DSS are outside the scope of this review.

In this section I target the analysis of clinicians' compliance with CIG as well as CIG versioning, which is needed in order to evolve CIGs when compliance analysis indicates that they are not being followed enough. Version management is also necessary when CIG versions evolve as new CPG versions become available and also when locally-adapted CIG versions are created.

4.7.1. CIG maintenance: compliance analysis

In this section, I review methods for measuring compliance to CIGs, machine learning methods for mining actual process paths from event logs, and visualization of process compliance results.

4.7.1.1. Measuring compliance with CIG recommendations. Compliance to guideline recommended actions can be measured by studying log files of CIG execution engines combined with data from electronic health records that document patient's clinical state, as measured by history and physical findings and laboratory and imaging test results, as well as medications prescribed and medical procedures ordered. Yet medical records are often incomplete and do not record all actions done by physicians and patients, resulting in compliance scores that are lower than the actual compliance. It has been shown that patient self-reporting can improve compliance scores [84] due to a variety of reasons. In particular, many patients were misidentified as being overdue for health maintenance procedures when they had obtained them in other places.

A methodology for evaluating compliance with CIG actions was developed by van de Klundert et al. [85]. They developed and tested dynamic programming formulations for compliance measurement in clinical pathways, based on partially ordered data in medical records and pathway definitions. Their pathway compliance measurement methods cope with the dynamics and flexibility of pathways, recognizing the fact that deviations are common to practice and are not necessarily negative. By developing an integral numerical compliance measure, the models and methods allow scoring deviations at various severity levels. The authors have applied their model to real-life data from the years 2001–2005 at the Maastricht University Medical Centre.

A concept related to compliance is critiquing [86]. Critiquing compares clinical actions performed by a physician with a predefined set of actions, such as those specified in a CIG. In case differences exist, the critiquing system provides insight into the extent to which they are compatible. Groot and colleagues [86] used mod-

---

[1] A trial in clinical settings of the PROforma RetroGram DSS for HIV dosing based on genotype demonstrated virological outcome improved by 33%. See http://www.cossac.org/files/EBGP06.pdf.

el checking (see Section 4.4.2.2) for critiquing based on a clinical guideline for breast cancer. In their approach, the ideal actions are given by a formal model of a clinical guideline, and the actual actions are derived from real world patient data. They start with a clinical guideline represented as an Asbru CIG and translate it to a state transition system. Non-compliance patterns are formulated in computation tree logic (CTL). Model checking was used to investigate whether a part of the actual treatment is consistent with the guideline.

In the works of Klundert et al. and Groot et al., the actual clinical actions performed by a physician are compared with a predefined set of actions recommended by the CIG.

In contrast, Advani et al. [87] developed an approach that measures compliance based on the *intentions* of the guideline, which as in the Asbru language, are temporal constraints on patient outcome states or required physician actions. The rationale is that deviations from guidelines are often necessary due to many reasons. For example, due to the fact that guidelines are incomplete and do not cover all possible comorbidities, physicians can recognize a comorbidity in their patient that has not been addressed by the guideline or find that the patient cannot cope with side effects of the preferred treatment. Even lack of organizational resources could necessitate deviation from guideline recommendations [88]. In this case, physicians could deviate from the recommended action while still following the guideline's higher-level intentions (e.g., to reduce blood pressure to a certain goal level within a specified time frame) and offer an acceptable alternative that complies with the guideline's intentions. Advani et al. introduce the Quality Indicator Language (QUIL) [87] and a scoring methodology that may be used to formally specify quality indicators as temporal constraints on physician behavior and patient outcomes derived from clinical guidelines. An action constraint may be specified as a temporal pattern of regular daily therapeutic doses of a drug in the beta-blocker class that is given continuously for six months. The methodology for quality assessment based on QUIL has been implemented within the MedCritic system, which was designed to work with EON- and Asbru-based DSSs. Note however, that unlike the approach used by Klundert et al. [85], QUIL does not allow modeling the (partial) order in pathway structures and patient record structures.

While compliance could be evaluated in real time while the CIG is being executed, due to the reasons outlined above, non-compliance could be falsely detected. This could happen even if reasoning at higher-level intentions is implemented, due to incompleteness of medical records and of the knowledge contained in clinical guidelines. Therefore many of the systems for detecting non-compliance operate retrospectively and CIG execution engines allow physicians to override system recommendations without implementing critiquing services. However, to understand better reasons for non-compliance, CIG-based DSS such as NewGUIDE include user interfaces (called Reasoning on Medical Actions – RoMA) for justifying non-compliance [88].

4.7.1.2. Process mining methods. Process mining methods analyze process logs to discover the actual process model behind them or compare the actual process model to the recommended clinical algorithm or CIG, a technique known as delta analysis [89]. Such comparison can bring insight into the most commonly followed pathways and their deviation from the recommended practice, which can help organizations in improving the design of their healthcare processes. In addition, the discovered process model can also be used to predict the paths for new patients. They are most often used in the business process management (BPM) community but are starting to gain increasing interest in the medical informatics community because of that community's interest in

addressing flexible processes that are so common in healthcare. Instead of log files, time stamped EHR data reflecting healthcare activities can serve as a source of mining data.

One of the earliest papers from this category is that by Lin et al. from 2001 [90]. The authors developed a data mining method that discovers the time-dependency pattern of clinical activity pathways that occur in practice with high frequency. Their method starts with data preprocessing (including pruning out activities that are not meaningful) then proceeds to mining time-dependency patterns. In the resulting activity graphs, they report for each activity in the graph the mean start and end times with their standard deviation. The mined patterns are then used to predict how new patients should be assigned to clinical pathways. In the prediction step, an association technique is used to evaluate the relationship between the diagnosis data and treatment paths in order to predict the paths for new patients. The authors evaluated their methodology with clinical pathway data of stroke patients.

More recently, two papers discussing process mining were published in the medical informatics journals reviewed, both from the research group of Huang et al. [91,92]. The authors use two different mining approaches in those papers. In [91] they develop a sequence pattern mining algorithm, related to the approach of Lin et al. [90] and in [92] they present a dynamic-programming algorithm related to Klundert et al. [85]. The authors explain that those two related works are "confined to one or several well-structured fragments of patient-linked treatment processes" while they can find frequent and meaningful medical behaviors in more complex care pathways. Most process mining methods developed by the BPM community do not try to detect the critical activities and do not analyze the time span of activities but only their order. In addition, the high flexibility of healthcare processes often results in complicated spaghetti-like structures produced by these mining algorithms. Huang et al. try to address these shortcomings.

Similar to Lin et al. [90], Huang et al. [91] discover critical medical patterns of ordered activities and provides knowledge about quantified temporal relations. They call these patterns "closed clinical pathway patterns". A closed clinical pathway pattern is a sequence of activities with specified temporal constraints that are found in high frequency and is minimal in the sense that there is no super-pattern with the same support for the closed clinical pathway pattern.

While in [91] Huang et al. discovered many small-sized clinical patterns, in their paper [92] they focused on discovering a concise and comprehensive summary that describes the entire structure of a clinical pathway, while revealing essential/critical medical behaviors in specific time intervals over the whole time period of the pathway. Similar to the approach used by Klundert and coauthors [85], the authors use a dynamic-programming algorithm to solve the summarization problem in polynomial time. The pathway is summarized by segmenting the observed time period of the pathway into continuous and overlapping time intervals, and discovering frequent medical behavior patterns in each specific time interval from the log.

The two approaches of Huang et al. were evaluated via real-world data-sets, which were extracted from Zhejiang Huzhou Central hospital of China with regard to six specific diseases, i.e., bronchial lung cancer, gastric cancer, cerebral hemorrhage, breast cancer, infarction, and colon cancer, in two years.

4.7.1.3. Visualization of CIG compliance results. In addition to mining techniques, visualization is key for presenting mining results to researchers and clinicians in order to explore how contextual information impacts performed treatment steps. Visual exploration is important for finding reasons and explanations for observed phenomena in the measured patient data. CareVis [93] is an

exploration tool that provides an integrated visualization of computerized protocols and temporal patient data. CareVis provides multiple simultaneous coupled views of the complex underlying data structure of treatment plans and patient data. The views are based on the concepts of clinical algorithm maps and LifeLines which are well-known to domain experts, which have been extended to support plan representation in the Asbru CIG language.

4.7.2. CIG maintenance: versioning and evolution

Because CIGs evolve over time and are also adapted to local settings, multiple CIG versions need to be managed. Naturally, support should be provided to the process of adaptation of CIGs to the evolving knowledge and local settings but at the same time, it should be possible to manage patients who have started their treatment process according to previous versions of CIGs and determine the stage at which they can migrate to the revised CIG. In addition, due to legal questions, retrospectively answering queries regarding the CIG knowledge that was available at an earlier time point and its application to patient's clinical data should be supported. The support of multi-version guidelines is addressed in the paper by Grandi et al. [94]. The authors propose techniques to model and to provide efficient personalized access to very large collections of multi-version clinical guidelines, which can be stored both in textual and in executable format in an XML repository. Efficiency is achieved by specifying for knowledge chunks their version information and by storing all knowledge in the same knowledge base, such that when desired, knowledge chunks that satisfy the patient's profile and context information can be assembled together to create a patient-specific plan. Version information in the knowledge base relates to (1) temporal characteristics, such as validity, efficacy, transaction, availability, proposal, approval, and publication times, and (2) clinical context information arranged in hierarchies relating to any medical concept, for example, diseases or demographic data, which the knowledge chunks include as indications for recommendations. In this way, patient profile and context information can be used by an automated personalization service to efficiently build on-demand a guideline version tailored to a specific patient profile and time stamp. When generating the view, sections that are relevant include those that relate to specific concepts that characterize the patient, and additional relevant sections are those that relate to super-classes of the concepts that characterize the patient. Relating to time, a version can be created for the current time stamp, assembling latest available knowledge, or for a past time stamp – to reconstruct a previous guideline version for retrospective analysis. As the authors point out, their approach can be considered as a compatible extension of DeGeL [78], which supports multi-version representation capabilities and efficient personalization query facilities.

Other researchers considered other aspects that are important for supporting the adaptation process. Focusing on knowledge acquisition and evolution, Kaiser et al. developed the information extraction LASSIE tool [47], mentioned in Section 4.2.3, which extracts clinical actions that can be marked up using the DELT/A tool guideline/CIG markup tool. LASSIE supports guideline evolution by identifying the parts of a new CPG that did not change from a previous version. If these parts have been previously marked up and formalized, then this markup is retained. In addition, information extracted from the new parts of the guideline is highlighted.

4.8. CIG sharing

A very large effort goes into the disambiguation and formalization of narrative clinical guidelines as CIGs. This work could be leveraged if CIGs, or in general, computer-interpretable knowledge,

could be shared among different implementing institutions, to the benefit of the community. In fact, this was the motivation of the InterMed Collaboratory for developing the GLIF CIG formalism, as reflected in the titles of the papers reviewed in Section 4.1 [23–25]. Section 4.5.1 presented the related approach by Wang [74] where an execution engine was developed that can execute different clinical guidelines, hence achieving sharing at the execution level without the necessity to adopt a single CIG formalism as a standard representation. While almost all of these papers are already included in other CIG themes, I decided to devote a separate section to sharability of CIGs due to the importance of this topic and to the fact that it is an integral part of the life-cycle of CIG development allowing implementing institutions to start with a structured CIG rather than from an unstructured narrative CPG (see Fig. 1).

Over the years, several approaches for sharing have been considered. One approach that was initially considered by the Inter-Med Collaboratory was to develop GLIF as an interchange between different CIG formalisms. But this approach was found to be unfeasible due to the many unique features that each CIG formalism employs. A second approach of establishing a single standard CIG-formalism was also not well-accepted by the community. Hence, the focus shifted to finding ways to share structured, computer-interpretable CIG knowledge. The GELLO object-oriented guideline expression language [33] reviewed in Section 4.1.3 and the Virtual Medical Record [53] information model reviewed in Section 4.3.1 were developed for sharing purposes and in fact became HL7 standards.

Boxwala et al. [95] propose structuring and sharing decision-support knowledge that that has not been adapted to a local institution from the original CPG and is free from proprietary implementation details. The addition of structure to narrative guidelines is done gradually until a formal executable level of specification is achieved; in this sense it is similar to the approaches of markup used in BRIDGE-Wiz, GEM-cutter, the consensus markup-method of DeGeL, LASSIE, and Stepper, discussed in Section 4.2. In Boxwala's methodology, the first layer of structure that is added to the unstructured narrative guideline is the semi-structured representation of individual recommendations that are not arranged into a clinical algorithm. Each recommendation is modeled as a decision about the interventions (actions) that are possible in a specified clinical scenario (patient context), which can include term definitions. The next layer is a structured representation that is computable and precise yet is independent of a particular DSS implementation and of the workflow in a particular clinical setting. In this layer, scenarios are defined as logical expressions, written in GELLO (see below). Action definitions include decision alternatives and factors that influence the selection of an alternative. The object model underlying the patient data in logical expressions is adapted from the Clinical Statements model from the Health Information Technology Standards Panel's Continuity of Care Document Patient Summary Construct (C32) specification [96] and uses HL7 data types. The objects can reference standard terminology codes. The executable specification layer is structured according to specifics of the clinical decision support system (e.g., CIG formalism) and adapted to a specific clinical information system in an implementing institution. Therefore, it is less sharable. A complementary approach in which general clinical decision support rules can be incrementally customized to take into account setting-specific factors is examined in the United States Office of the National Coordinator for Health IT (ONC) SHARPC 2B project (http://www.uthouston.edu/nccd/projects/sharpc/index.htm) which examines adaptive decision support.

In the past two or three years, sharing computer-interpretable knowledge has been addressed in many panels in prominent con-
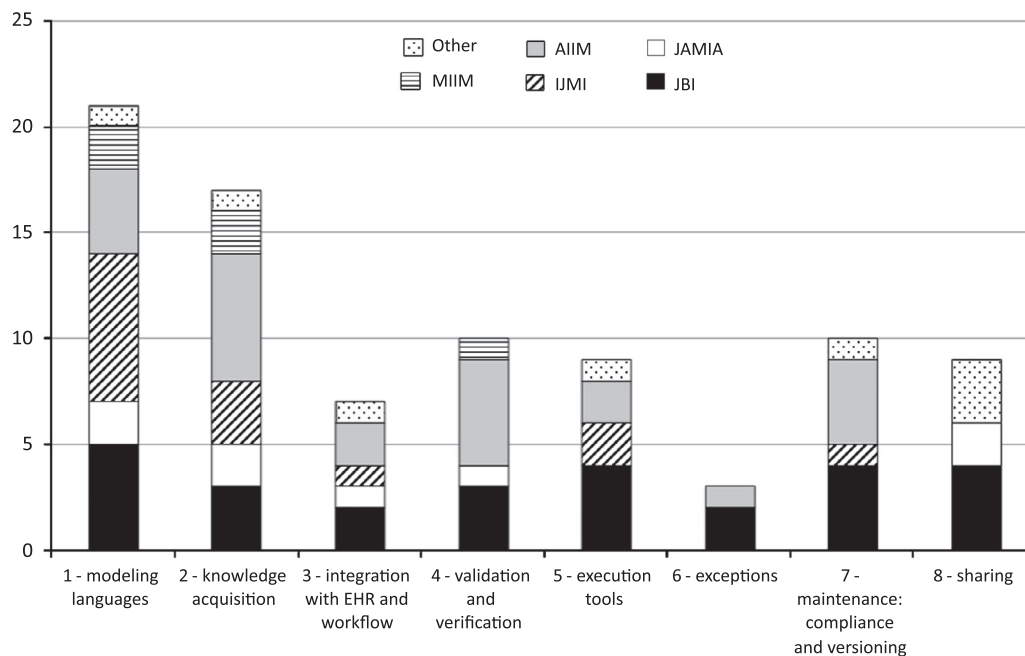
Fig. 2. Number of papers published in each category from each journal.

ferences and is the topic of research of several US national health IT projects. As an example, the Medinfo 2010 panel, "Sharing guideline knowledge: can the dream come true?" (http://www.open-clinical.org/sharingGLknowledge.html) discussed efforts made in this direction, highlighting technical, content-wise, and business model considerations. Some of the technical aspects include the size of knowledge chunks that is useful for sharing. As in [13], many researchers are acknowledging that sharing small chunks of decision logic such as if-then rules may allow more reuse and easier knowledge maintenance than sharing larger pieces of guideline knowledge. Other important issues to guideline sharing include accelerating the development or licensing of required, pragmatic standards and addressing medico-legal liability concerns [97]. To facilitate sharing of CIGs, different organizations such as the DeGel project (see Section 4.5.2), openclinical.org and the Morningside initiative [98] are setting up libraries of CIGs and executable components in different guideline representation formalisms.

5. Discussion

In this paper, CIG-related papers published in JBI were manually classified into themes that span the entire life-cycle of CIG development process. Additional papers from other leading journals and several seminal papers were then classified into these themes. Section 5.1 addresses the appropriateness and limitations of the classification categories and some of the emerging CIG research themes not covered by it. In Section 5.2 I examine the trends in CIG-related research.

5.1. Appropriateness of the classification scheme used in this paper

The appropriateness of the classification scheme can be evaluated with respect to existing methodology and review papers related to CIGs as well as to new emergent topics of research. Several questions can be asked:

5.1.1. Are there any CIG-related topics that were not major themes of JBI papers?

To answer this question, I considered the CIG-related topics addressed in other journals covered in this review as well as recent review papers related to CIGs. In general, JBI papers cover most themes covered by papers published in other prominent journals (see how JBI papers represent a prominent proportion of papers for each of the eight categories in Table 2 and Fig. 2).

The categories used in this paper are also in agreement with current review papers that address CIGs. The review paper by Gooch and Roudsari [99] addressed implementation challenges for process-oriented health information systems. The categories discovered by the classification algorithm used in that paper covered topics similar to those used in this paper, including knowledge acquisition, guideline formalization as a CIG, local workflow integration, verification, validation, and execution. But in addition, other categories from that paper, although addressed in this review were not prominent categories here, including: overcoming implementation barriers, modeling clinical goals, and temporal expressions.

The review paper from Hasman's [3] group focused on factors that influence the design and implementation of guidelines. The themes addressed in this review paper fit well with this paper's categorization and includes: assessing the quality of guideline development and implementation (related to knowledge acquisition), formalization methods to translate paper guidelines into CIGs, models for CIGs that facilitate their design, effects of guidelines with respect to patient outcome and efficiency of care, and barriers to guideline compliance.

In addition to these recent reviews, three early review papers addressed the modeling features of CIG modeling languages. Wang's [100] review has focused on guideline representation primitives, process models, and their relationship to a patient's clinical status. The comparison by Peleg and colleagues [2] identified eight dimensions that capture the structure of CIGs, falling into two broad categories – structuring guidelines as plans for decisions and actions, and linking the guideline to patient data and medical concepts. De Clercq's [20] review addresses, in addition to guideline representation issues, aspects concerning guideline acquisi-

tion, verification and execution. The paper by Mulyar et al. [71] compared CIG modeling methods in terms of their support of workflow patterns. Finally et al. [72] review computer-based execution of clinical guidelines.

In addition to the reviews mentioned above, which appeared in journals, three books addressing CIGs have been published following three international workshops that were devoted specifically to CIGs [101–103].

5.1.2. What are some of the new emerging CIG-related themes?

By following the new ongoing projects of leaders in CIG research, I can see several topics of research that are being currently addressed. Some were already introduced in this review and include CIG knowledge sharing. Other emerging topics include:

(a) CIG interaction and basic medical knowledge: one unsolved research question is how can we concurrently apply multiple CIGs to patients that have comorbid clinical conditions? This topic is addressed in paper [69] included in Section 4.4.2.3 on CIG verification. Yet there have not been enough work published (especially in the journals reviewed here) to warrant a separate theme for this important topic. A related question is how to exploit basic medical knowledge [61] in order to adapt CIG recommendations to the specific patient case at hand, for example, if the patient has other comorbidities or if the local settings does cannot accommodate a certain medical procedure.

Still, recent research projects are currently addressing this topic. One example is the K4CARE project (http://www.k4car-e.net) that resulted in an ontology that is used for personalization of health-care knowledge to support clinical decisions for chronically ill patients with comorbid conditions [104]. The ontology formally represents the health-care concepts related to the care of chronically ill patients at home, including syndromes, diseases, social issues, signs and symptoms, assessments, and interventions, and the relationships and constraints between them. The knowledge sources for the ontology included CPGs and scientific papers related to a selected set of diseases and syndromes. Each disease, syndrome and social issue in the ontology is related to a formal intervention plan that represents the corresponding treatment as a state-decision–action (or SDA) diagram, which is a CIG formalism. Based on the ontology and the patient's EHR data, several personalization steps are carried out. They include (i) personalizing the patient condition – starting with confirmed data about the patient (PatientCase), the Background class (i.e., basic medical knowledge from the ontology) containing concepts in the case profile that should also be observed for the patient is used to help physicians to define a more accurate profile of the patient's condition; (ii) personalizing the involved actors based on organizational knowledge and data regarding the patient's care providers; and (iii) personalizing intervention plans by first generating individual intervention plans corresponding to the diseases of that patient and then integrating unified intervention plans for the comorbid clinical conditions.

Fuzzy Cognitive Maps [13], discussed in Section 4.1.2, could in theory be the basis for combining knowledge originating in several CPGs in order to address patients with comorbidities.

Another project addressing CIG interaction is the GuideLine INteraction Detection Assistant (GLINDA) project (https://bmir-gforge.stanford.edu/gf/project/glinda/).

The two following challenges are addressed in the European FP7 MobiGuide project: Guiding Patients Anytime Everywhere (www.mobiguide-project.eu)

(b) Patient-centric CIG-based DSSs: how does the patients' personal context [105], their preferences, and the system's technological state and quality of data affect decision making? How can decision-theoretic models be used to support shared decision-making by patients and care providers?

(c) Developing ubiquitous CIG-based guidance systems that use service-oriented architecture (SOA) to provide decision-support in event-driven and synchronous modes through web-based and smartphone user interfaces. In MobiGuide, decision-support is distributed between a backend CIG-based DSS Server and a light-weight DSS that runs on a Smartphone. The light-weight system uses a simpler projection of the CIG knowledge and has access to patient biosignal data collected by wearable sensors that are connected to the smartphone. The mobile DSS can provide decision-support even if there is no connectivity to the backend DSS Server. SOA architectures are also being explored by the Health Level 7 Clinical Decision Support Work Group [106].

(d) Developing process *learning* methods that mine relationships between process context (i.e., patient's clinical parameters at process entry time and events that are external to the process but influence the patient's state), pathways taken, and outcomes achieved. Such analysis could suggest process improvement. Initial work has been done in this direction includes the Learning Process Model (LPM) suggested in [107]. LPM recognizes that for certain specific context, deviation from the recommended pathways may sometimes improve outcomes for that context. LPM operates by first discovering relevant context groups where following a certain path results in a specific outcome. Then it discovers the decision rules that govern activity selection (pathways) for reaching good outcomes. The data requirements needed to apply the context discovery algorithm has been applied to acute care processes (e.g., a urinary tract infection treatment process) has been discusses in [108].

5.1.3. Limitations of this review

This review included papers from five prominent medical informatics journals, assuming that important work are eventually published in these journals. However, related research originating in communities other than the medical informatics community is not reviewed in this paper. Examples include works from the business process management sub-community called ProHealth – process support for healthcare, which have been holding workshops annual workshops since 2007. Such work includes the rich process mining work of van der Aalst [89] and the work of Dadam and Reichert on the Application Development based on Encapsulated Premodeled Process Templates (ADEPT) system [109], related to CIG maintenance. ADEPT provides change operations for dynamically revising process model (i.e., CIG) instances that are being executed for a patient, and for revising process schemas. Interestingly, the development of ADEPT started after Dadam and Reichert have applied workflow technology at the Women's Hospital of the University of Ulm and were challenged by the need to provide flexible yet robust and easy-to-use process support that is so inherent to the healthcare domain.

A related limitation is that the methodology of the review precludes some important and relevant works that were too recent to be published as journal papers. Research work tends to be published first in workshops and conferences before it appears in prominent journals. As a result, there are many recent papers related to CIGs that are not covered in this review. They include, among others papers related to mapping of CIG knowledge to EHR data [110], CIG execution engines [111,112], exception handling [113], and process mining [114,107,108].
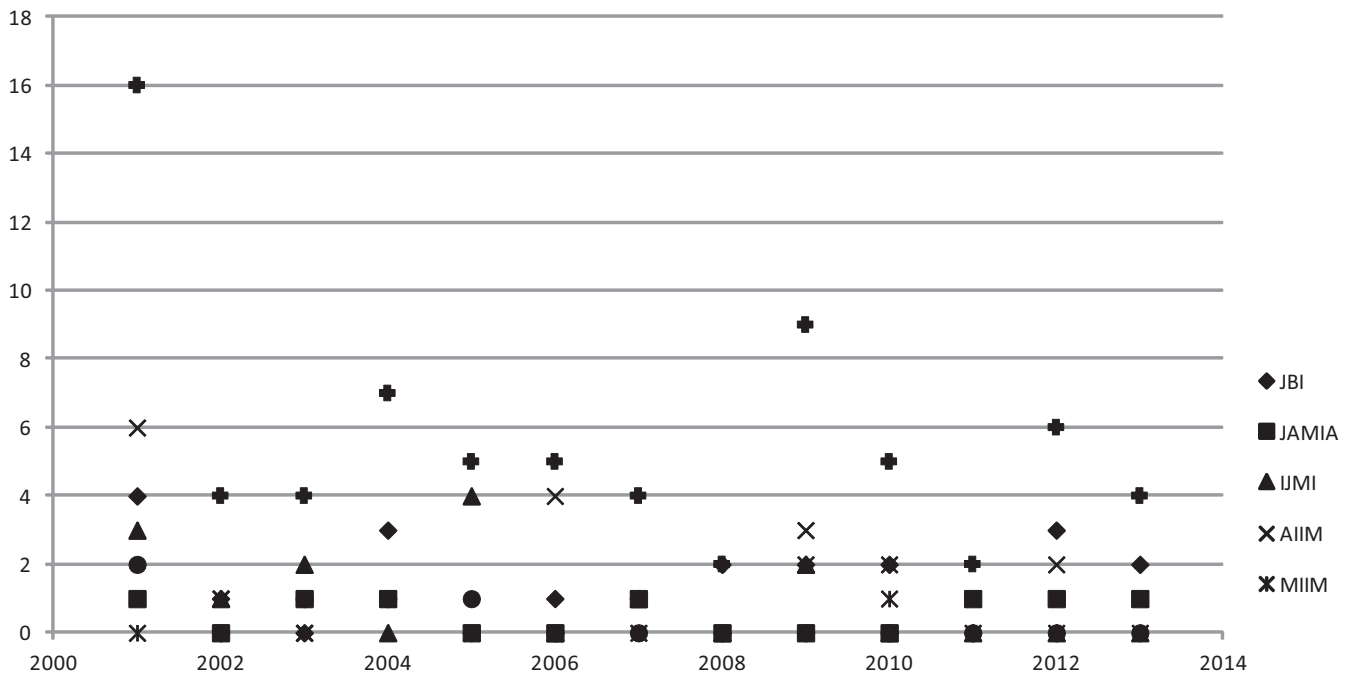
Fig. 3. The number of publications in each journal and in total, as a function of time.

Lastly, although the choice of search keywords was wide, some CIG-related papers published in the five journals reviewed could also have been missed. One noted case is that of the paper by Riaño [104] and colleagues discussed above, which is related to multiple CIGs applied to comorbid patients. This paper was not retrieved as it uses other keywords (ontologies, medical decision support systems, and health care personalization). Another noted paper is that of German et al. [54], related to mapping of a DSS' knowledge base (which can be a CIG knowledge based) to electronic medical records.

Another noted case is the paper from Terenziani's group [115], which belongs to the topic of CIG maintenance (Section 4.7) yet did not meet the search criteria. The authors propose a model for update of shared knowledge or data and exemplify their approach for versioning of CIG knowledge. Their proposal, General Proposal Vetting Model (GPVM), is a general model to cope with change proposal evaluation and commitment (vetting) in relational temporal databases, where the knowledge is stored and maintained. The bi-temporal model (i.e., supports valid time and transaction time) allows analysis of physicians' compliance to the CIG knowledge used at the time that they were treating the patient. In the vetting process, proposers propose updates to the CIG knowledge and evaluators incrementally accept or reject them. One of the main contributions of this approach is the definition of a general and principled treatment of alternative proposals in relational databases.

5.2. Trends in CIG-related research

Fig. 2 shows the number of papers published in each category from each journal. Observing Fig. 1 and Table 2, one can see that the themes identified are quite prominent, each including at least two papers from JBI and always some other papers from other prominent journals. Some of the themes include at least 10 papers each: CIG modeling languages (21), knowledge acquisition (17), execution tools (10), and validation and verification (10), while other themes show opportunity and need for further research

(e.g., exception handling in CIGs (three papers whose earliest paper is from 2009).

Please note that some papers appear in more than one theme. As noted in Section 4.8, this is most significant in theme #8 CIG sharing, but some overlap occurs in a few other places (see paper numbers in bold that appear in more than one theme in Table 2).

Observing Table 2, we can also compare the total number of papers presenting new methods related to CIGs published in the five journals reviewed since 2001. We can see that most papers were published in Artificial Intelligence in Medicine (22) and JBI (21). A medium number (13) of papers was published in International Journal of Medical Informatics but only 7 and 5 papers were published in JAMIA and in Methods of Information in Medicine. Because CIG representation is usually done through ontologies and other artificial intelligence representation methods, Artificial Intelligence in Medicine is not surprisingly a valuable source of methodological papers regarding CIGs. The high number of methodological papers published in JBI is in line with this journals aims and scope (http://cdn.elsevier.com/promis_misc/YJBIN_Aims-Scope.pdf): "JBI is intended to complement rather than to compete with the other major journals in biomedical informatics. In particular, we wish to emphasize papers the elucidate methodologies that generalize across biomedical domains and that help to form the scientific basis for the field. Papers will tend to be concerned with information technology rather than medical devices, and on underlying methods rather than system descriptions or summative evaluations."

The picture would have looked differently if I had reviewed also review papers and evaluation or application papers, as JAMIA is a journal that specializes in such a topic.

Fig. 3 shows the number of publications in each journal and in total (marked by a '+' symbol), as a function of time. The total number of papers about CIGs in the time period reviewed (2001–February 7, 2013) is between 2 and 16 papers with an average of 5.75 papers a year (for years 2001–2012) with significantly more papers in the year 2001(16 papers). This high number of papers may be partly due to the Workflow Management and Clinical Guidelines special issue that was published in Arti-

ficial intelligence in Medicine in 2001, in which four CIG related papers (all papers in that issue) were published. But mostly it seems to suggest that the highest interest in CIGs was in the late 1990's until 2001. Nevertheless, this research topic is continuing to be an important topic with many open challenges as reflected by the fact that although 2013 has just started, four published or accepted papers have already been devoted to CIG-related methods.

Acknowledgment

References

[1] Institute of Medicine. Clinical practice guidelines we can trust. Consensus report, 2013. <http://www.iom.edu/Reports/2011/Clinical-Practice-Guidelines-We-Can-Trust.aspx> [accessed on 23.04.13].
[2] Peleg M, Tu SW, Bury J, Ciccarese P, Fox J, Greenes RA, et al. Comparing computer-interpretable guideline models: a case-study approach. J Am Med Inform Assoc 2003;10(1):52–68.
[3] Latoszek-Berendsen A, Tange H, van den Herik HJ, Hasman A. From clinical practice guidelines to computer-interpretable guidelines. A literature overview. Methods Inf Med 2010;49(6):550–70.
[4] Hripcsak G, Ludemann P, Pryor TA, Wigertz OB, Clayton PD. Rationale for the Arden syntax. Comput Biomed Res 1994;27(4):291–324.
[5] Koutkias V, Kilintzis V, Stalidis G, Lazou K, Niès J, Durand-Texte L, et al. Knowledge engineering for adverse drug event prevention: on the design and development of a uniform, contextualized and sustainable knowledge-based framework. J Biomed Inform 2012;45(3):495–506.
[6] Serban R, ten Teije A, van Harmelen F, Marcos M, Polo-Conde C. Extraction and use of linguistic patterns for modelling medical guidelines. Artif Intell Med 2007;39(2):137–49.
[7] 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(5):488–98.
[8] Georg G, Seroussi B, Bouaud J. Extending the GEM model to support knowledge extraction from textual guidelines. Int J Med Inform 2005;74(2–4):79–87.
[9] Colombet I, Aguirre-Junco A-R, Zunino S, Jaulent M-C, Leneveut L, Chatellier G. Electronic implementation of guidelines in the EsPeR system: a knowledge specification method. Int J Med Inform 2005;74(7–8):597–604.
[10] Seroussi B, Bouaud J, Antoine E-C. ONCODOC: a successful experiment of computer-supported guideline development and implementation in the treatment of breast cancer. Artif Intell Med 2001;22(1):43–64.
[11] Seroussi B, Bouaud J, Chatellier G. Guideline-based modeling of therapeutic strategies in the special case of chronic diseases. Intl J Med Inform 2005;74(2–4):89–99.
[12] Sanders GD, Nease RF, Owens DK. Design and pilot evaluation of a system to develop computer-based site-specific practice guidelines from decision models. Med Decis Mak 2000;20(2):145–59.
[13] Papageorgiou EI, Roo JD, Huszka C, Colaert D. Formalization of treatment guidelines using Fuzzy Cognitive Maps and semantic web tools. J Biomed Inform 2012;45(1):45–60.
[14] Shahar Y. The "Human Cli-Knowme" project: building a universal, formal, procedural and declarative clinical knowledge base, for the automation of therapy and research. In: Lect Not Artif Intell 6924. Springer; 2012. p. 1–22.
[15] Musen MA, Tu SW, Das AK, Shahar Y. EON: a component-based approach to automation of protocol-directed therapy. J Am Med Inform Assoc 1996;3(6):367–88.
[16] 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(1–2):29–51.
[17] Seyfang A, Miksch S, Marcos M. Combining diagnosis and treatment using Asbru. Int J Med Inform 2002;68(1–3):49–57.
[18] Latoszek-Berendsen A, Talmon J, de Clercq P, Hasman A. With good intentions. Int J Med Inform 2007;76(Suppl. 3):S440–6.
[19] Latoszek-Berendsen A, de Clercq P, van den Herik J, Hasman A. Intention-based expressions in GASTINE. Methods Inform Med 2009;48(4):391–6.
[20] de-Clercq PA, Blom JA, Korsten HH, Hasman A. Approaches for creating computer-interpretable guidelines that facilitate decision support. Artif Intell Med 2004;31(1):1–27.
[21] Anselma L, Terenziani P, Montani S, Bottrighi A. Towards a comprehensive treatment of repetitions, periodicity and temporal constraints in clinical guidelines. Artif Intell Med 2006;38(2):171–95.
[22] Terenziani P, Molino G, Torchio M. A modular approach for representing and executing clinical guidelines. Artif Intell Med 2001;23(3):249–76.
[23] Boxwala AA, Tu S, Zeng Q, Peleg M, Ogunyemi O, Greenes RA, et al. Towards a representation format for sharable clinical guidelines. J Biomed Inform 2001;34:157–69.
[24] 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(3):170–81.
[25] Boxwala AA, Peleg M, Tu S, Ogunyemi O, Zeng Q, Wang D, et al. GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines. J Biomed Inform 2004;37(3):147–61.
[26] Skonetzki A, Gausepohl HJ, van der Haak M, Knaebel S, Linderkamp O, Wetter T. HELEN, a modular framework for representing and implementing clinical practice guidelines. Methods Inf Med 2004;43(4):413–26.
[27] Panzarasa S, Madde S, Quaglini S, Pistarini C, Stefanelli M. Evidence-based careflow management systems: the case of post-stroke rehabilitation. J Biomed Inform 2002;35(2):123–39.
[28] Quaglini S, Stefanelli M, Lanzola G, Caporusso V, Panzarasa S. Flexible guideline-based patient careflow systems. Artif Intell Med 2001;22:65–80.
[29] Ciccarese P, Caffi E, Quaglini S, Stefanelli M. Architectures and tools for innovative health information systems: the guide project. Int J Med Inform 2005;74(7–8):553–62.
[30] Sutton DR, Fox J. The syntax and semantics of the PROforma guideline modeling language. J Am Med Inform Assoc 2003;10(5):433–43.
[31] Tu SW, Campbell JR, Glasgow J, Nyman MA, McClure R, McClay J, et al. The SAGE guideline model : achievements and overview. J Am Med Inform Assoc 2007;14(5):589–98.
[32] Schadow G, Russler DC, McDonald CJ. Conceptual alignment of electronic health record data with guideline and workflow knowledge. Intl J Med Inform 2001;64(2–3):259–74.
[33] Sordo M, Boxwala AA, Ogunyemi O, Greenes RA. Description and status update on GELLO: a proposed standardized object-oriented expression language for clinical decision support. In: Medinfo, 2004. p. 164–8.
[34] Patel VL, Arocha JF, Diermeier M, Shortliffe EH, Greenes RA. Methods of cognitive analysis to support the design and evaluation of biomedical systems: the case of clinical practice guidelines. J Biomed Inform 2001;34(1):52–66.
[35] Peleg M, Gutnik LA, Snow V, Patel VL. Interpreting procedures from descriptive guidelines. J Biomed Inform 2006;39(2):184–95.
[36] Stausberg J, Bilir H, Waydhas C, Ruchholtz S. Guideline validation in multiple trauma care through business process modeling. Int J Med Inform 2003;70(2–3):301–7.
[37] Patel VL, Allen VG, Arocha JF, Shortliffe EH. Representing a clinical guideline in GLIF: individual and collaborative expertise. J Am Med Inform Assoc 1998;5(5):467–83.
[38] Ohno-Machado L, Gennari JH, Murphy S, Jain NL, Tu SW, Oliver DE, et al. The guideline interchange format: a model for representing guidelines. J Am Med Inform Assoc 1998;5(4):357–72.
[39] Patel VL, Branch T, Wang D, Peleg M, Boxwala AA. Analysis of the process of encoding guidelines: a comparison of GLIF2 and GLIF3. Methods Inform Med 2002;41(2):102–13.
[40] Shalom E, Bar G, Yarkoni A, Young O, Martins SB, Vaszar L, et al. A quantitative assessment of a methodology for collaborative specification and evaluation of clinical guidelines. J Biomed Inform 2008;41(6):889–903.
[41] Shiffman RN, Michel G, Rosenfeld RM, Davidson C. Building better guidelines with BRIDGE-Wiz: development and evaluation of a software assistant to promote clarity, transparency, and implementability. J Am Med Inform Assoc 2012;19(1):94–101.
[42] Shiffman RN, Dixon J, Brandt C, Essaihi A, Hsiao A, Michel G, et al. The guideline implementability appraisal (GLIA): development of an instrument to identify obstacles to guideline implementation. BMC Med Inform Decis Mak 2005;5:23.
[43] Shiffman RN, Michel G, Essaihi A, Thornquist E. Bridging the guideline implementation gap: a systematic, document-centered approach to guideline implementation. J Am Med Inform Assoc 2004;11(5):418–26.
[44] Svátek V, Růzicka M. Step-by-step mark-up of medical guideline documents. Intl J Med Inform 2003;70(2–3):329–35.
[45] Peleg M, Tu SW. Design patterns for clinical guidelines. Artif Intel Med 2009;47(1):1–24.
[46] Serban R, ten Teije A. Exploiting thesauri knowledge in medical guideline formalization. Methods Inform Med 2009;48(5):468–74.
[47] Kaiser K, Miksch S. Versioning computer-interpretable guidelines: semi-automatic modeling of "living guidelines" using an information extraction method. Artif Intell Med 2009;46(1):55–66.
[48] Kaiser K, Akkayaa C, Miksch S. How can information extraction ease formalizing treatment processes in clinical practice guidelines? A method and its evaluation. Artif Intell Med 2010;39(2):151–63.
[49] Kosara R, Miksch S. Metaphors of movement: a visualization and user interface for time-oriented, skeletal plans. Artif Intell Med 2001;22(2):11–131.
[50] Goud R, Hasman A, Strijbis A-M, Peek N. A parallel guideline development and formalization strategy to improve the quality of clinical practice guidelines. Int J Med Inform 2009;78(8):513–20.
[51] Bates DW, Kuperman GJ, Wang S, Gandhi T, Kittler A, Volk L, et al. Ten commandments for effective clinical decision support: making the practice of evidence-based medicine a reality. J Am Med Inform Assoc 2003;10(6):523–30.

[52] Peleg M, Keren S, Denekamp Y. Mapping computerized clinical guidelines to electronic medical records: knowledge-data ontological mapper (KDOM). J Biomed Inform 2008;41(1):180–201.

[53] Johnson PD, Tu SW, Musen MA, Purves I. A virtual medical record for guideline-based decision support. Proc AMIA Symp 2001:294–8.

[54] German E, Leibowitz A, Shahar Y. An architecture for linking medical decision-support applications to clinical databases and its evaluation. J Biomed Inform 2009;42(2):203–18.

[55] González-Ferrer A, ten Teije A, Fdez-Olivares J, Milian K. Automated generation of patient-tailored electronic care pathways by translating computer-interpretable guidelines into hierarchical task networks. Artif Intell Med. 2013;57(2):91–109.

[56] Boehm BW. Software risk management. IEEE Computer Society Press; 1989.

[57] Peleg M, Shachak A, Wang D, Karnieli E. Using multi-perspective methodologies to study user interactions with the front-end of a guideline-based decision-support system for diabetic-foot care. Intl J Med Inform 2009;78(7):482–93.

[58] Miller PL, Frawley SJ, Sayward FG. Maintaining and incrementally revalidating a computer-based clinical guideline: a case study. J Biomed Inform. 2001;34(2):99–111.

[59] Choi J, Currie LM, Wang D, Bakken S. Encoding a clinical practice guideline using guideline interchange format: a case study of a depression screening and management guideline. Intl J Med Inform 2007;76(Suppl. 2). S302–7.

[60] Miller PL. Domain-constrained generation of clinical condition sets to help test computer-based clinical guidelines. J Am Med Inform Assoc 2001;8(2):131–45.

[61] Bottrighi A, Chesani F, Mello P, Montali M, Montani S, Terenziani P. Conformance checking of executed clinical guidelines in presence of basic medical knowledge. In: Business process management workshops, 2011. p. 200–11.

[62] Duftschmid G, Miksch S. Knowledge-based verification of clinical guidelines by detection of anomalies. Artif Intell Med 2001;22(1):23–41.

[63] Duftschmid G, Miksch S, Gall W. Verification of temporal scheduling constraints in clinical practice guidelines. Artif Intell Med 2002;25(2): 93–121.

[64] Pérez B, Porres I. Authoring and verification of clinical guidelines: a model driven approach. J Biomed Inform 2010;43(4):520–36.

[65] Domínguez E, Pérez B, Zapata M. Towards a traceable clinical guidelines application. A model-driven approach. Methods Inform Med 2010;49(6): 571–80.

[66] Hommersom A, Groot P, Balser M, Lucas P. Chapter 4. Formal methods for verification of clinical practice guidelines. In: ten Teije A, Miksch S, Lucas PJF, editors. Computer-based medical guidelines and protocols: a primer and current trends. Studies in health technology and informatics, vol. 139. IOS Press; 2008. p. 63–80.

[67] Bottrighi A, Giordano L, Molino G, Montani S, Terenziani P, Torchio M. Adopting model checking techniques for clinical guidelines verification. Artif Intell in Med 2010;48(1):1–19.

[68] ten Teije A, Marcos M, Balser M, van Croonenborg J, Duelli C, van Harmelen F, et al. Improving medical protocols by formal methods. Artif Intell Med 2006;36(3):193–209.

[69] Wilk S, Michalowski W, Michalowski M, Farion K, Hing MM, Mohapatra S. Mitigation of adverse interactions in pairs of clinical practice guidelines using constraint logic programming. J Biomed Inform 2013;46(2):341–53.

[70] Grando MA, Glasspool D, Fox J. A formal approach to the analysis of clinical computer-interpretable guideline modeling languages. Artif Intell Med 2012;54(1):1–13.

[71] Mulyar N, van der Aalst WMP, Peleg M. A pattern-based analysis of clinical computer-interpretable guideline modeling languages. J Am Med Inform Assoc 2007;14(6):781–7.

[72] Isern D, Moreno A. Computer-based execution of clinical guidelines: a review. Int J Med Inform 2008;77(12):787–808.

[73] Wang D, Peleg M, Tu SW, Boxwala AA, Ogunyemi O, Zeng Q, et al. Design and implementation of the GLIF3 guideline execution engine. J Biomed Inform 2004;37(5):305–18.

[74] Wang D, Peleg M, Bu D, Cantor M, Landesberg G, Lunenfeld E, et al. GESDOR – a generic execution model for sharing of computer-interpretable clinical practice guidelines. In: Proc AMIA Symp, 2003. p. 694–8.

[75] Young O, Shahar Y, Liel Y, Lunenfeld E, Bar G, Shalom E, et al. Runtime application of Hybrid-Asbru clinical guidelines. J Biomed Inform 2007;40(5):507–26.

[76] de Clercq PA, Hasman A, Blom JA, Korsten HH. The application of ontologies and problem-solving methods for the development of shareable guidelines. Artif Intell Med 2001;22(1):1–22.

[77] 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 Inform 2001;64(2–3):285–318.

[78] Shahar Y, Young O, Shalom E, Galperin M, Mayaffit A, Moskovitch R, et al. A framework for a distributed, hybrid, multiple-ontology clinical guideline library, and automated guideline-support tools. J Biomed Inform 2004;37(5):325–44.

[79] Moskovitch R, Shahar Y. Vaidurya: a multiple-ontology, concept-based, context-sensitive clinical-guideline search engine. J Biomed Inform 2009;42(1):11–21.

[80] Moskovitch R, Cohen-Kashi S, Dror U, Levy I, Maimon A, Shahar Y. Multiple hierarchical classification of free-text clinical guidelines. Artif Intell Med 2006;37(3):177–90.

[81] Grando A, Peleg M, Glasspool D. A goal-oriented framework for specifying clinical guidelines and handling medical errors. J Biomed Inform 2009;43(2):287–99.

[82] Grando AM, Peleg M, Cuggia M, Glasspool D. Patterns for collaborative work in health care teams. Artif Intell Med 2011;53(3):139–60.

[83] Grando MA, Glasspool D, Boxwala A. Argumentation logic for the flexible enactment of goal-based medical guidelines. J Biomed Inform 2012;45(5): 938–49.

[84] Staroselsky M, Volk La, Tsurikova R, Pizziferri L, Lippincott M, Wald J, et al. Improving electronic health record (EHR) accuracy and increasing compliance with health maintenance clinical guidelines through patient access and input. Int J Med Inform 2005;75(10–11):693–700.

[85] van de Klundert J, Gorissen P, Zeemering S. Measuring clinical pathway adherence. J Biomed Inform 2010;43(6):861–72.

[86] Groot P, Hommersom A, Lucas PJ, Merk RJ, ten Teije A, van Harmelen F, et al. Using model checking for critiquing based on clinical guidelines. Artif Intell Med 2009;46(1):19–36.

[87] Advani A, Shahar Y, Musen M. Medical quality assessment by scoring adherence to guideline intentions. In: AMIA symp, 2001. p. 2–6.

[88] Quaglini S. Chapter 9. Compliance with Clinical Practice Guidelines. In: ten Teije A, Miksch S, Lucas PJF, editors. Computer-based medical guidelines and protocols: a primer and current trends. Studies in health technology and informatics, vol. 139. IOS Press; 2008. p. 160–79.

[89] van der Aalst WMP. Process mining: discovery, conformance and enhancement of business processes. Springer; 2011.

[90] Lin F, Chou S, Pan S, Chen Y. Mining time dependency patterns in clinical pathways. Int J Med Inform 2001;62(1):11–25.

[91] Huang Z, Lu X, Duan H. On mining clinical pathway patterns from medical behaviors. Artif Intell Med 2012;56(1):35–50.

[92] Huang Z, Lu X, Duan H, Fan W. Summarizing clinical pathways from event logs. J Biomed Inform 2012;46(1):111–27.

[93] Aigner W, Miksch S. CareVis: integrated visualization of computerized protocols and temporal patient data. Artif Intell Med 2006;37(3):203–18.

[94] Grandi F, Mandreoli F, Martoglia R. Efficient management of multi-version clinical guidelines. J Biomed Inform 2012;45(6):1120–36.

[95] Boxwala A, Rocha BH, Maviglia S, Kashyap V, Meltzer S, Kim J, et al. A multi-layered framework for disseminating knowledge for computer-based decision support. J Am Med Inform Assoc 2011;18(Suppl. 1). i132–9.

[96] Healthcare information technology standards panel. C 32 – HITSP summary documents using HL7 continuity of care document (CCD) component. <http://www.hitsp.org/ConstructSet_Details.aspx?&PrefixAlpha=4&PrefixNumeric=32>.

[97] Kawamoto K, Hongsermeier T, Wright A, Lewis J, Bell DS, Middleton B. Key principles for a national clinical decision support knowledge sharing framework: synthesis of insights from leading subject matter experts. J Am Med Inform Assoc 2013;20(1):199–207.

[98] Morningside Group. Collaborative development of a knowledge management repository to accelerate adoption of clinical decision support, 2007. <http://www.amia.org/public-policy/reports-and-fact-sheets/morningside-initiative-concept-paper>.

[99] Gooch P, Roudsari A. Computerization of workflows, guidelines, and care pathways: a review of implementation challenges for process-oriented health information systems. J Am Med Inform Assoc 2011;18(6):738–48.

[100] Wang D, Peleg M, Tu SW, Boxwal a AA, Greenes RA, Patel VL, et al. Representation primitives, process models and patient data in computer-interpretable clinical practice guidelines: a literature review of guideline representation models. Intl J Med Inform. 2002;68(1-3):59–70.

[101] Heller B, Löffler M, Musen M, Stefanelli M, editors. Computer-Based Support for Clinical Guidelines and Protocols Editors. In: Studies in health technology and informatics, vol. 83, 2001.

[102] Kaiser K, Miksch S, Tu SW, editors. Computer-based support for clinical guidelines and protocols. In: Proceedings of the symposium on computerized guidelines and protocols (CPG 2004). Studies in health technology and informatics, vol. 101, 2004.

[103] ten Teije A, Miksch S, Lucas P. Computer-based medical guidelines and protocols: a primer and current trends. Studies in health technology and informatics, vol. 139. IOS Press; 2008.

[104] Riaño D, Real F, López-Vallverdú JA, Campana F, Ercolani S, Mecocci P, et al. An ontology-based personalization of health-care knowledge to support clinical decisions for chronically ill patients. J Biomed Inform 2012;45(3): 429–46.

[105] Fux A, Peleg M, Soffer P. How can personal context improve adherence to clinical guidelines. In: Prohealth/KR4HC. Tallinn, Estonia, September 3, 2012.

[106] Kawamoto K, Honey A, Rubin K. The HL7-OMG healthcare services specification project: motivation, methodology, and deliverables for enabling a semantically interoperable service-oriented architecture for healthcare. J Am Med Inform Assoc 2009;16(6):874–81.

[107] Soffer P, Ghattas J, Peleg M. A goal-based approach for learning in business processes. In: Salinesi C, Souveyet C, Ralyte J, editors. Intentional perspectives on information systems engineering. Springer; 2010. p. 239–56. ISBN-10: 3642125433 ISBN-13: 978–3642125430.

[108] Ghattas J, Peleg M, Soffer P. Data requirements for process learning. Int J Knowl Org 2013;3(1) [special issue on Process support in healthcare – Part II].

[109] Reichert M, Dadam P. ADEPT flex – supporting dynamic changes of workflows without losing control. J Intell Inform Sys. 1997;10(2):93–129.

[110] Marcos M, Maldonado JA, Moner D, Bosc D. An archetype-based solution for the interoperability of computerised guidelines and electronic health records. In: AIME. 2011. p. 276–85.

[111] Le XH, Luque A, Wang D. Development of guideline-driven mobile applications for clinical education and decision support with customization to individual patient cases. In: Proc AMIA annual symp, 2012. p. 14642.

[112] Jafarpour B, Abidi SR, Sibte S, Abidi R. Exploiting OWL reasoning services to execute ontologically-modeled clinical practice guidelines. In: AIME. LNCS, vol. 6747; 2011. p. 307–11.

[113] Leonardi G, Bottrighi A, Galliani G, Terenziani P, Messina A, Francesco Della Corte. Exceptions handling within GLARE clinical guideline framework. In: Proceedings of AMIA, symposium, 2012. p. 512–21.

[114] Mans R, Schonenberg H, Leonardi G, Panzarasa S, Cavallini A, Quaglini S, et al. Process mining techniques: an application to stroke care. In: Medical informatics, Europe, 2008. p. 573–8.

[115] Anselma L, Bottrighi A, Montani S, Terenziani P. Managing proposals and evaluations of updates to medical knowledge: theory and applications. J Biomed Inform 2013;46(2):363–76.