Computerization of Clinical Guidelines: an Application of Medical Document Processing

Gersende Georg

SPIM ; Inserm ERM 202, 15 rue de l'École de Médecine, F-75006 Paris, France

1 Introduction: Clinical Guidelines as Normalized Documents

Clinical Guidelines are being developed as a tool to promote Best Practice in Medicine. They are usually defined as "systematically developed statements to assist practitioner and patient decisions about appropriate Healthcare for specific clinical circumstances" *(Institute of Medicine, 1990)*. The Institute's Committee on Practice Guidelines further clarified this definition by specifying *appropriate care* as: "the expected health benefit exceeds the expected negative consequences by a sufficient margin that the care is worth providing".

Clinical Guidelines bridge the gap between clinical science and practice. Their first concern is the growing awareness of large variations in clinical practice [1]. The second is the fact that Health professionals have difficulties in keeping up-to-date with the overwhelming volume of new scientific evidence for good clinical practice [2]. Another problem is the growing cost of Health services which prompts the need for Best Practice. A study conducted by the Juran Institute concluded that 30% of Health costs could be reduced and outcomes improved if quality issues were addressed [3]. As a result, Clinical Guidelines are touted as vehicles for improving the quality of Healthcare as well as decreasing costs.

Clinical Guidelines are also poised to play an important role in the diffusion and standardization of medical knowledge, as they rely on recent concepts of *Evidence-Based Medicine*. David Sackett defined *Evidence-Based Medicine* as "the conscientious, explicit and judicious use of current best evidence in making decisions about the care or individual patients. The practice of *Evidence-Based Medicine* means integrating individual clinical expertise with the best available clinical evidence from systematic research" [4]. However, we should emphasize that *Evidence-Based Medicine* is in no way downgrading medical standards to reduce costs. Indeed, in many cases, Clinical Guidelines incorporate the most current knowledge about Best Practice.

Recent studies have suggested that the diffusion of paper Guidelines only had a limited impact. Previous research on Clinical Guidelines concluded that their use was not widespread [5]. Many reasons have been put forward to account for this situation, among them the difficulty to use Clinical Guidelines (38% of respondents) [6]. Medin and Ross [7] hypothesize that this is due to the detailed processing required for written texts. However, another hypothesis invokes disagreements on the recommendations which may arise from the focus of the Guideline (*e.g.* the

kind of patients for whom the Guideline is intended may differ from the actual patient under consideration [8]).

This is why, the same studies have demonstrated that the inclusion of Clinical Guidelines within Decision Support Systems (DSS) has a significant potential to improve compliance of Health professionals. Several studies have proposed to evaluate the impact of DSS. One of them systematically reviewed feedback directed to Healthcare providers or patients [9]. It concluded that reminders are more effective than feedback in modifying physician behavior. Patient-directed reminders can improve medication adherence as well. In a similar fashion, Lobach and Hammond [10] showed that DSS based on Clinical Guidelines are an effective tool for assisting clinicians in the management of diabetic patients. The effect of computer-generated advice on clinician behavior was measured through the rate of compliance with Guideline recommendations. Compliance for the group receiving recommendations was 32.0% versus 15.6% for the control group. As a result, the use of DSS based on Clinical Guidelines may improve the quality of medical care.

2 The Authoring of Clinical Guidelines

Computerization of Clinical Guidelines is a major challenge for their successful dissemination. Because of their textual nature and their relation to standardized medical knowledge, their computerization involves many different fields of Health Informatics, from document processing to knowledge-based systems. In this chapter, we discuss the computerization of Clinical Guidelines through the various aspects of their life cycle, from production to consultation and use (Figure 1).

2.1 Problems Encountered with Clinical Guidelines Authoring

Some studies have noticed the lack of a standard structure in the publication of text-based Clinical Guidelines. Indeed, Clinical Guidelines were being "produced and disseminated by a variety of government and professional organizations and because these Guidelines are largely in narrative-form, they are sometimes ambiguous and generally lack the structure and internal consistency that would allow execution by computer" [11]. This is why several studies are focusing on authoring and standardization of Clinical Guidelines through their computerization. Recently, Shiffman et al. [12] concluded that one major factor affecting the quality of Clinical Guidelines is the fact that Guideline development panels are commonly composed of people who are inexperienced in Guideline authoring. This important point explains some of the standardization problems encountered. On the other hand, Guideline production could be "intentionally ambiguous to satisfy a consensus process used to create it, or the Guideline may lack coverage (incomplete cases) or appear to be contradictory" [13].

Clinical Guidelines may be complex, and often composed of elaborate collections of procedures containing logical gaps or contradictions that can generate

ambiguity. Many Clinical Guidelines also suffer from a complex structure, involving the nesting of procedures or complicated temporal or causal relations. An understanding of the semantics of Clinical Guidelines and of their potential interpretation by physicians may improve this, and ultimately the usability of Clinical Guidelines [8].

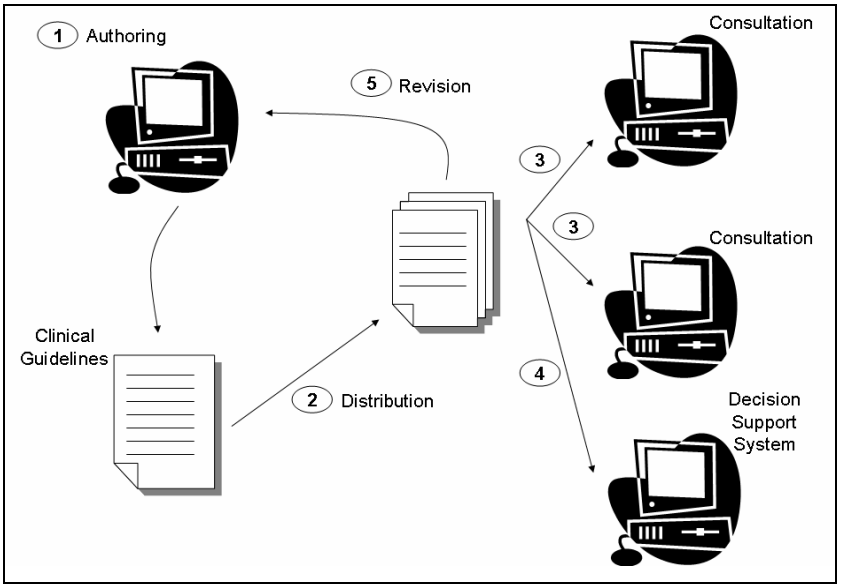

Fig. 1. The life cycle of Clinical Guidelines.

2.2 Clinical Guidelines' Structure

Clinical Guidelines focus on the management of specific diseases, such as diabetes or hypertension and are typically presented in textual and algorithmic forms. We can observe separate chapters for the different steps in disease management, for example diagnosis and therapeutic strategy (Figure 2).

Textual Guidelines consist of the description of a procedure, usually based on scientific evidence from clinical studies. To complement written Guidelines, algorithms (Figure 2) are often integrated into Clinical Guidelines that describe therapeutic or diagnostic procedures. Algorithmic representations organize all of the relevant information into a directly applicable form, and therefore may aid decision-making [14].

Clinical Guidelines can be characterized by their specific structure and their specific knowledge content. Their logical form has inspired much research in document processing, while their knowledge content has served as a starting point

for experiments with knowledge-based systems. There is no standardized language in which to encode the elements of a Clinical Guideline in a consistent way which would support its computerization. Indeed, as traditional forms of Clinical Guidelines may not make it immediately clear how to apply the Guideline [11], research into their structure is a precondition to their computerization.

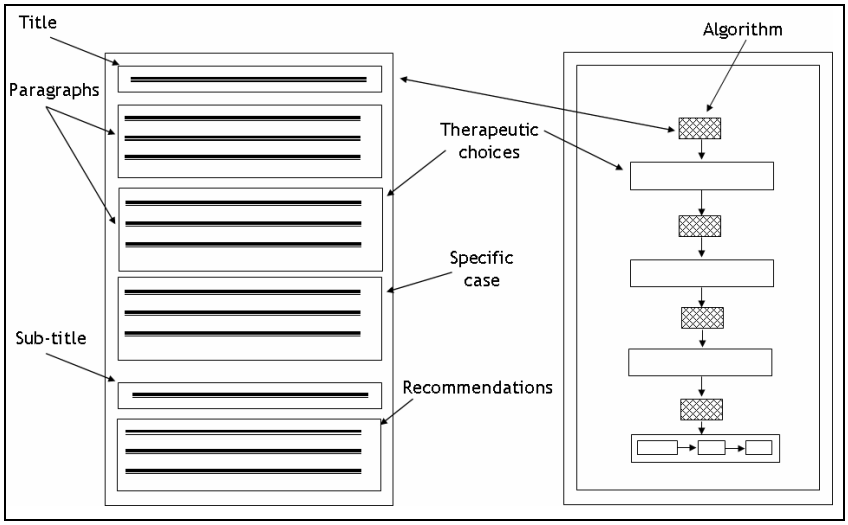

Fig. 2. Structure of Clinical Guidelines.

In this context, we present state-of-the-art research in document management techniques applied to Clinical Guidelines, and report recent results from our own work in the computerization of the Guideline production workflow.

The knowledge content of Clinical Guidelines can be characterized by two major dimensions. One consists in the representation of medical decision processes (for instance, the therapeutic strategy) and the other in the attempt at disseminating standardized clinical knowledge. Several authors have proposed knowledge representations for the decision process contained in Clinical Guidelines. For instance, the Guideline Interchange Format (GLIF) represents the decision processes as flowcharts [15], while the PRO*forma* model [16] enhances Guidelines' contents with task-based formalisms representing the decision processes as task models, while the GUIDE system [17] formalizes the decision process using Petri nets. Others researchers have proposed to relate the contents of Clinical Guidelines to Medical Logic Modules (MLM) embedding the standardized knowledge they convey [18].

3 Knowledge Representation of Clinical Guidelines

Several authors have proposed knowledge representations for the decision process contained in Clinical Guidelines. To highlight commonalities and differences between these approaches, we present them through a common example based on Chronic Cough management Guidelines, presented in a recent study [19]. This extract contains three recommendations:

(i) "Chronic cough is cough that lasts for at least 3 weeks."
(ii) "Chest radiographs should be ordered before any treatment is prescribed in nearly all patients with chronic cough (Grade II-2)."
(iii) "Chest radiographs do not have to be routinely obtained before beginning treatment, for presumed PNDS (post nasal drip syndrome) in young nonsmokers, in pregnant women, or before observing the result of discontinuation of an ACE-I (ACE Inhibitor) for 4 weeks for patients who developed cough shortly after beginning to take an ACE-I."

3.1 The Guideline Interchange Format (GLIF)

The GLIF formalism has been collaboratively developed by groups at Columbia, Stanford, and Harvard universities (working together as the InterMed Collaboratory). The main interest of GLIF derives from the importance of sharing Guidelines among different institutions and software systems. This model behaves as a meta-model for describing and representing the components of Clinical Guidelines.

3.1.1 Flowcharts' Customization and the Description of Decision Processes

The first version of GLIF [15] was published in 1998. GLIF supported Guideline modeling as a flowchart with a specific syntax. To tackle the complexity of Clinical Guidelines, the flowchart is extended to enable the specification of a Clinical Guideline as temporally ordered steps. The "Conditional Step" is the equivalent of the "if" part of the rule-based "if … then" structure. The "Action Step" is the equivalent of the "then" part of an "if … then" statement.

Concurrency was modeled using "Branch Steps" which specify multiple steps that can happen concurrently, and "Synchronization Steps" which synchronize the execution of the multiple branches.

3.1.2 Limitations of the GLIF Approach

Some GLIF studies [11] have found that the process of structuring Clinical Guidelines information into GLIF requires that users have extensive coding abilities (Figure 3). Without this, programmers often experience difficulties which result in substantial variability when encoding Clinical Guidelines into the GLIF format [11]. As a result, users require significant time to learn the GLIF language. Furthermore, studies have also found that by encoding a Clinical Guideline with

GLIF, the original integrity of the text-based Guideline can sometimes be lost [20].

A non-exhaustive list of shortcomings may be raised: (i) GLIF does not specify how to structure important attributes of Clinical Guidelines steps; (ii) the integration with heterogeneous clinical systems is difficult; (iii) its semantics is a mixture of concurrency and decision making; and (iv) important concepts are lacking, for example those for describing iteration, patient-state, exception conditions, and events.

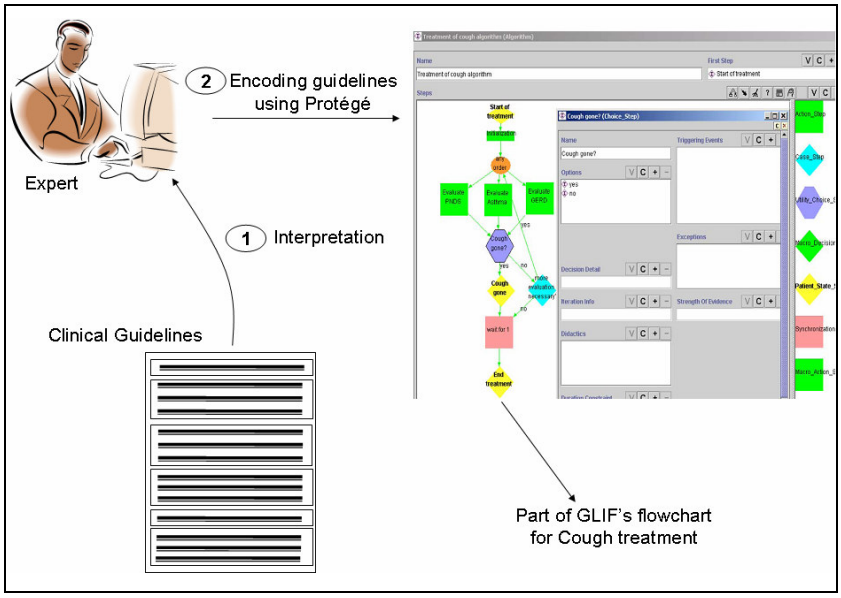

Fig. 3. Knowledge extraction from Clinical Guidelines using Protégé.

To tackle these problems, the latest release GLIF3 augments the GLIF2 specification to support versioning of GLIF-encoded Clinical Guidelines [21]. This is why three levels have been proposed to encode Clinical Guidelines with GLIF [22]: (1) The conceptual flowchart; (2) The computable specification (which is automatically checked for consistency); and (3) the implementable specification.

3.1.3 The Latest Release of GLIF: GLIF3

The Conceptual Flowchart

Flowcharts initially described in GLIF have been augmented using Unified Modeling Language (UML) class diagrams. This supports nesting mechanisms for representing complex Clinical Guidelines through iterative specification of Clinical Guidelines (for instance, through the nesting of Sub-Guidelines in action and deci-

sion steps). This provides a flexible decision model through a hierarchy of decision step classes (Figure 4).

For instance, using our standard example, Chronic Cough, which is shown in Figure 5 as an action step, can be expanded by zooming, through the nesting mechanism, to show its details in the form of yet another flowchart diagram. This decision hierarchy distinguishes between decision steps that can be automated (*case steps*) and those that have to be further specified by a physician (*choice steps*).

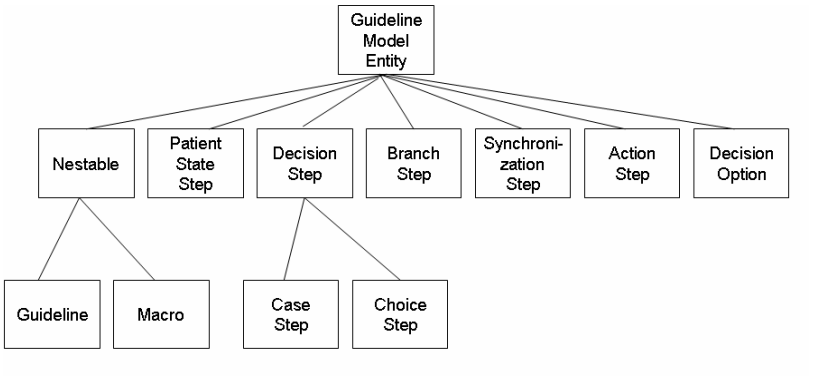

Fig. 4. Overview of the main classes in GLIF version 3.

The action specification model has been extended to include two types of actions: (1) Guideline-flow-relevant actions, such as invoking a Sub-Guideline, or computing values for clinical parameters; and (2) clinically relevant actions, such as issuing recommendations. Clinically relevant actions relate to the domain ontology such as prescriptions, laboratory tests, or referrals.

Finally, the branching and synchronization steps have been modified to remove redundancy in descriptions of parallel pathways in the Guideline flowchart.

Extending the Flowchart Representation

New representational features have been developed such as (i) describing *Iterations* and *Conditions* that control the iteration flow; (ii) describing *Events* and the triggering of actions by these events; (iii) describing *Exceptions* in Guidelines flow and associated exception-handling mechanisms; (iv) representing *Patient-State* as another Guideline step (a node in the flowchart). In this way, a *Patient-State Step* serves as an entry point into the Guideline.

The Computable Specification

The computable level stands between the abstract flowchart level (supported by GLIF2) and the implementation level (currently partially supported by GLIF3). The aim of the abstract flowchart level is to help authors and users visualize and understand the structure of Clinical Guidelines.

The Implementable Specification

A formal syntax for specifying expressions and criteria has been added to the model. It is based on a superset of the Arden Syntax logic grammar [22], and adds new operators such as "is a", "overlaps", "xor", "from now", "is unknown", and "at least *k* of …". This enables to describe a MLM (cf. section 'Arden Syntax') using a pattern of GLIF components which can be used to map GLIF encoded Clinical Guidelines into MLM.

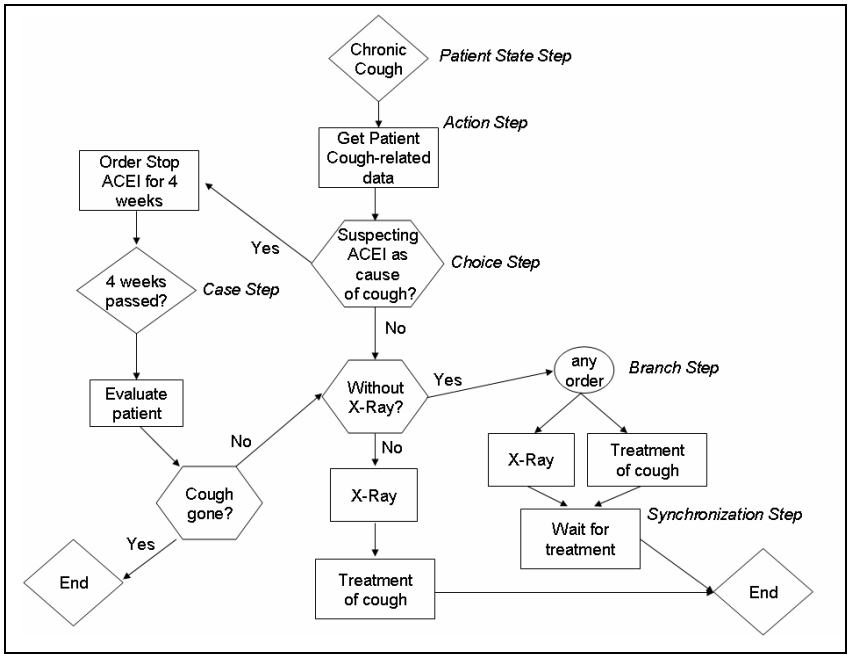

Fig. 5. Conceptual flowchart specification for part of the Chronic Cough Guideline.

GLIF3 is also based on a domain ontology that normalizes terms encoding. However, work on domain ontology in GLIF3 is still in progress. The proposed ontology is based on three layers. The *Core Layer* provides a standard interface to all medical data and concepts that may be represented and referenced by GLIF. The *Reference Information Model (RIM) Layer* provides a semantic hierarchy for medical concepts, and allows attribute specification for each class of medical data. The *Medical Knowledge Layer* contains a term dictionary (*e.g.*, UMLS) and can provide access to medical knowledge bases.

3.1.4 Current Research

Future versions of GLIF will explore structured representations for (1) specifying the various goals of each Guideline step, (2) incorporating probabilistic models for decision-making, and (3) incorporating patient preferences in each decision step.

Software tools are currently being developed for authoring, verifying, viewing, distributing, and executing Guidelines. The GLIF3's promoters are currently producing several examples Clinical Guidelines, described at these three levels, in order to evaluate GLIF3.

As a conclusion, this new version of GLIF combines the strengths of several knowledge representations, such as MLM, UML class diagram and ontologies. Standardization of Clinical Guidelines control-flow by HL7 also draws upon the GLIF model of linked Guideline steps. InterMed members are active participants of the HL7 Clinical Guidelines Special Interest Group and the Clinical Decision Support Technical Committee, thus contributing to the process of standardization of a shareable Guideline modeling language, which draws upon experiences from the GLIF project.

3.2 PRO*forma*

PRO*forma* is a formal knowledge representation language supported by acquisition and execution tools with the goal of supporting Guideline dissemination in the form of expert systems that assist patient care through active decision support and workflow management [16]. PRO*forma* was developed at the Advanced Computation Laboratory of Cancer Research, UK. The name PRO*forma* is a concatenation of the terms *proxy* ('authorized to act for another') and *formalize* ('give definite form to').

3.2.1 The Domino Autonomous Agent Model for Clinical Guidelines

PRO*forma* is an agent language which has developed out of research in logic-based decision making, workflow agents and process modeling [23]. The PRO*forma* model is based on the Domino autonomous agent model and includes several components such as: goals, situations (patient data), actions (clinical orders), candidate solutions, decisions (diagnosis, treatment), and plans (treatment and care plans).

PRO*forma* is based on predicate calculus augmented by non-standard logics. It has also a well-structured syntax, defined in a BNF-equivalent normal form for which applications can be developed using ordinary text editors and compilers. As a consequence, it combines logic programming and object-oriented modeling and is formally grounded in the R²L language which extends the standard syntax and semantics of Prolog with agent-like capabilities, including decisions, plans and actions, temporal reasoning, and constraints.

3.2.2 Encoding Clinical Guidelines in PRO*forma*

Like the GLIF model, PRO*forma* represents Guidelines as a directed graph in which the nodes are instances of a closed set of classes, called the PRO*forma* task ontology. Each Guideline in PRO*forma* is modeled as a plan consisting of a sequence of tasks.

The PRO*forma* task model (Figure 6) divides from the generic task (keystone) into four types: action, plan, decision, and enquiry. "Action" is a procedure to be carried out. This procedure may include presenting information or instructions to the user, sending email reminders, or calling other computer programs. "Plan" is the basic building block of Clinical Guidelines and may contain any number of tasks of any type, including other plans.

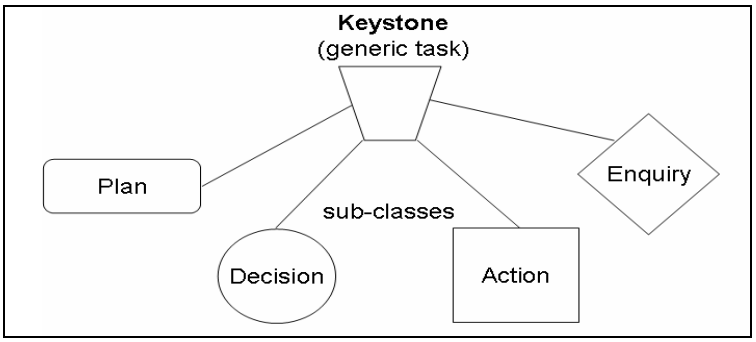

Fig. 6. Overall view of the PRO*forma* model.

"Decision" is a task presenting an option, *e.g.* whether to treat a patient or carry out further investigations, and "Enquiry" is a request for further information or data, required before proceeding with the application of the Clinical Guideline. All tasks share attributes describing goals, control flow, preconditions, and post-conditions. The simple task ontology should make it easier to demonstrate soundness and to teach the language to encoders.

3.2.3 The Graphical Editor "Arezzo"

The development environment for the Domino model consists of a graphical editor supporting the authoring process. An engine has also been developed to execute the Clinical Guideline specification as shown in Figure 7.

The "Composer" is used to create Guidelines, protocols and care pathways. The editor supports the construction of Clinical Guidelines in terms of the four task types described above. In the editor, logical and temporal relationships between tasks are captured naturally by linking them as required with arrows. Any procedural and medical knowledge required by the Guideline as a whole (or by an individual task) is entered using templates attached to each task. For each task, a current task state is defined, for example *Dormant* means that the task has not yet been started; *Requested* when an "Enquiry" task is activated and data values are sought for source data items. This is the default state for all tasks before the Guideline is started. All tasks and data items have attributes that define their properties and determine the behavior of the Guideline during enactment by the Arezzo Performer engine. The resulting instantiated graphical structure, as shown in Figure 8, is automatically converted into a database ready for execution.

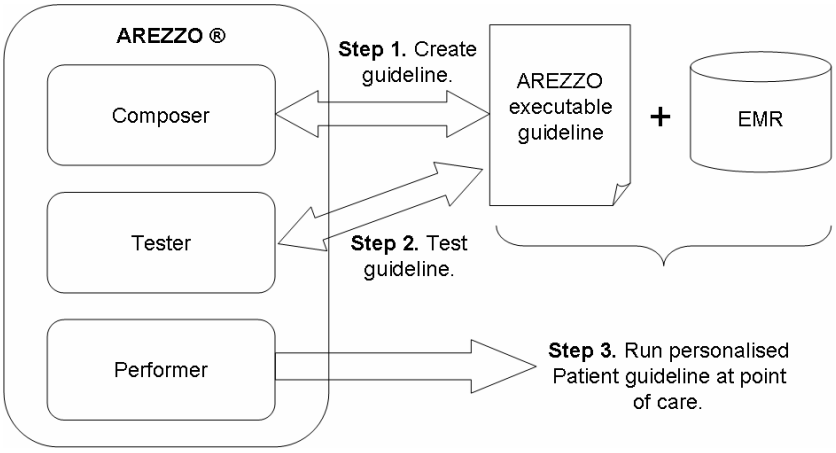

Fig. 7. Overview of Arezzo[1].

As an example, the top-level Cough Guideline is shown on the top left of Figure 8. The two inserts show nesting of the two plans of the top-level Guideline. The "Enquiry" task *(Initial assessment)* might request data about a patient's clinical signs. After this information is received, the "Action" *(Guideline is not appropriate)* or the "Plan" task *(CRX and initial treatment)* are handled. If the Clinical Guideline is appropriate, the "Plan" task is processed and the "Decision" task processes the data values.

Depending on which option the user chooses, the Guideline presents the appropriate "Action" task: *CRX first* or *CRX in parallel*. The scheduling constraint in the top-level Guideline states explains why the component *Investigations* will not be executed until the component *CRX and initial treatment* has been completed.

After the creation of the Guideline with the "Composer", the "Tester" (Figure 7) is used to test the Guideline logic before deployment. The engine can also be used as a tester during the application development phase. The "Performer" inference engine can then run the Guideline when making clinical decisions about a patient. This module also allows Guidelines to be embedded in existing Healthcare systems, linking seamlessly with local Electronic Medical Records and other applications to provide patient-specific assistance at the point of care.

As a conclusion, PRO*forma* offers a declarative interchange format for describing Clinical Guidelines together with a knowledge acquisition methodology and a tool set to simplify the composition and formal verification of applications.

PRO*forma* is the only approach which makes a distinction between a declarative language (*e.g.*, R²L), used during the Guideline acquisition phase and a procedural language (*e.g.*, L_{R^2L}) that is processed by a general interpreter in an execution engine [23]. All other approaches require a custom-developed execution

[1] http://www.infermed.com/arezzo/arezzo-components

engine, in which the different procedural aspects of the Guideline are encoded automatically (*e.g.*, a number of Java or C procedures that each executes a certain primitive).

In a similar fashion, tools of this kind can be used to assist normalization bodies in overseeing the preparation of Clinical Guidelines and protocols.

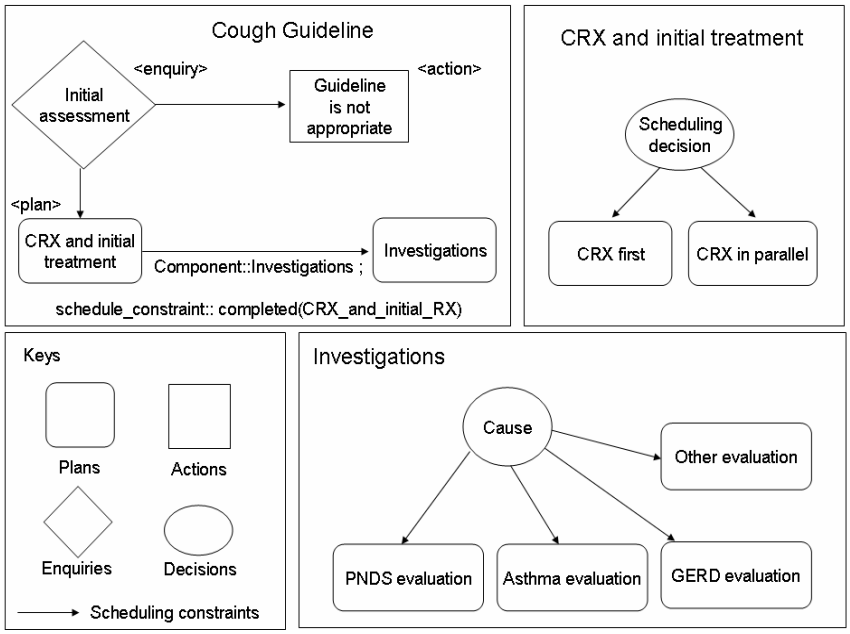

Fig. 8. Visualization of the Cough Guideline using Arezzo.

3.3 GUIDE

GUIDE is part of a Guideline modeling and execution framework being developed at the University of Pavia [17]. GUIDE aims to provide an integrated medical knowledge management infrastructure through a workflow called *careflow*.

3.3.1 Petri Nets for Representing Clinical Guidelines

GUIDE is based on Petri nets, a traditional formalism for modeling concurrent processes. The strength of the formalism, when applied to Healthcare, is its ability to support the modeling of complex concurrent processes (sequential, parallel and iterative logic flows) which are often part of Clinical Guidelines (see above).

In GUIDE, Petri nets have been extended to support the modeling of time, data and hierarchies. Clinical Guidelines could be represented directly through a Petri

net, using a graphical editor which enables non-expert users to build a Petri net (Figure 9). Considering the same example Clinical Guideline (Cough management) we can observe that it consists of a succession of ordered steps. Each step is triggered whenever the previous step is validated, and parallel or simultaneous steps can be represented corresponding to the temporal aspects of Clinical Guidelines. Petri net's computational properties have been largely used for workflow simulation, hence their use for simulating patients' management.

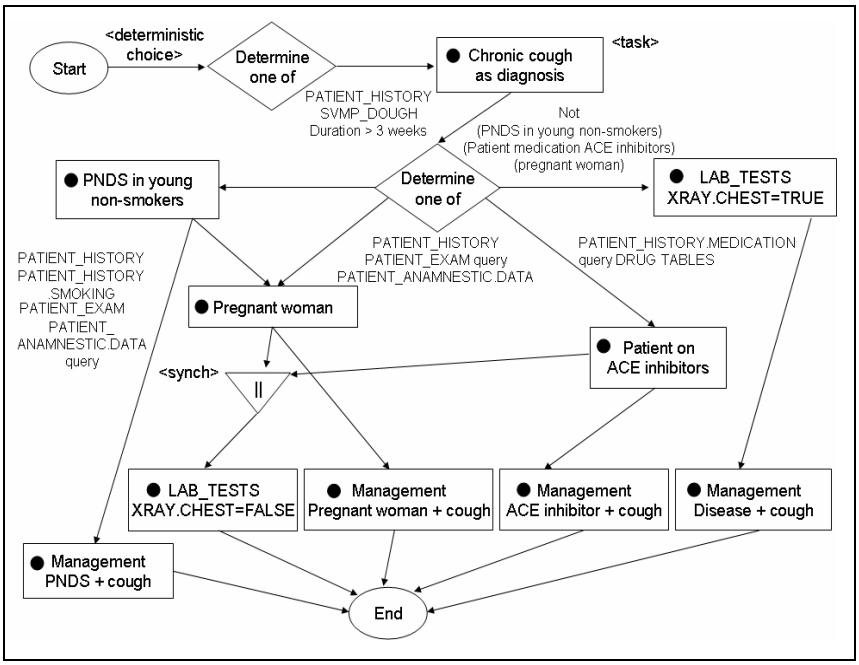

Fig. 9. The top-level GUIDE model of the Cough Guideline.

3.3.2 GUIDE's Workflow: the *Careflow* Approach

GUIDE is integrated into a workflow management system which is "a system that completely defines, manages, and executes workflow processes through execution of software whose order of execution is driven by a computer representation of the workflow process logic". It fully implements a Clinical Guideline and controls both its execution and outcome. The Workflow Management Coalition defines a workflow as: "The automation of a business process, in a whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules".

This definition can be transferred to the case of Clinical Guidelines. The promoters of GUIDE have coined the term *careflow* to refer to the workflow in the

context of patient care. Quaglini et al. [17] have proposed a methodology for integrating knowledge representation tools with commercial workflow tools. The commercial tools are Income™ and Oracle Workflow™, used for *careflow* model simulation and *careflow* implementation, respectively.

GUIDE is thus an intermediate step, oriented towards medical experts, by means of which Clinical Guidelines may be formalized. From the Petri net objects, a translation into the Workflow Process Definition Language (WPDL which is the language recommended by the Workflow Management Coalition) is obtained automatically. They adopt this standard representation in order to be able to exploit different existing products for the subsequent phases of *careflow* implementation.

3.3.3 Clinical Guidelines Encoded into the *Careflow*

Figure 10 shows the main methodological steps to build a Guideline-based *careflow* system. *Medical* and *Organizational knowledge* are combined and used to build a computational model of the Clinical Guideline implementation within a clinical environment.

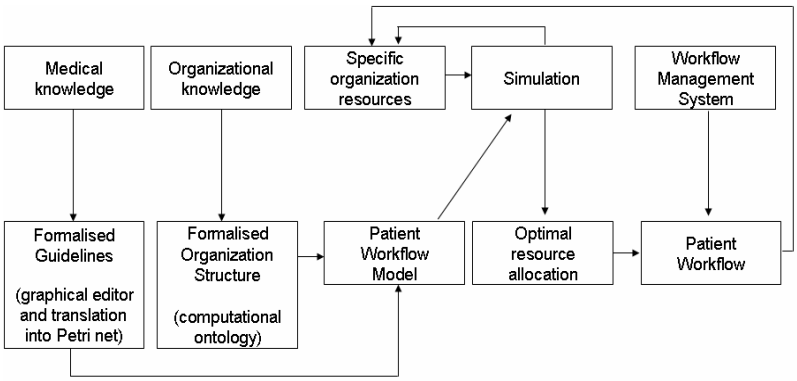

Fig. 10. A methodology for building a *careflow* management system.

Figure 10 above shows different kinds of feedback: the *Optimal resource allocation* is achieved by simulating different *Optimal resource allocations*, each arranged according to the previous simulation results.

GUIDE may also call external modules representing decisions as decision trees or influence diagrams that may take into account patient preferences, organizational preferences, and economic evaluations.

3.3.4 A New Approach for GUIDE

Recently, the GUIDE environment has integrated three main modules: the Guideline Management System (providing clinical decision support), the Healthcare Record System (providing access to patient data), and the Workflow Management

System or Careflow Management System (providing organizational support) as shown in Figure 11 [24].

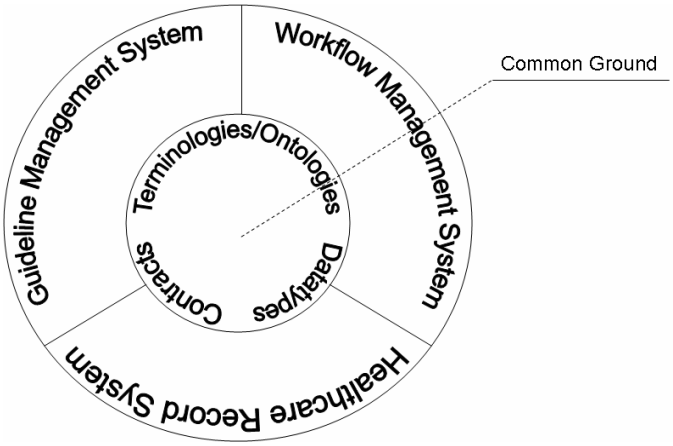

Fig. 11. The Pie model represents the separation of the three main modules.

This new approach is based on the separation between medical and organizational issues. Clinical Guidelines can be executed in three ways: (1) GUIDE can run a Clinical Guideline in simulation mode by simulating patient data; (2) it can simulate the effects of implementing a Clinical Guideline at a facility by translating a Clinical Guideline into Petri nets using the GUIDE editor, and augmenting the model with a knowledge base reflecting the facility's organizational structure; and (3) it can drive resource allocation and task management in clinical settings by using the Oracle™ Workflow tool.

A given Healthcare organization could perform some local adaptation of the GUIDE formalized Guideline. Revision could be performed both at local level and at global level with a feedback to the original Guideline authors in order to improve the original model. Several projects are in progress following this approach [24].

4 Disseminating Standardized Clinical Knowledge

The first approach for computerizing Clinical Guidelines has relied on rule-based systems. However, the developers who first implemented Clinical Guidelines found that many "published Guidelines suffer from unclear definition of specifications, incompleteness, and inconsistency, and these deficiencies compromise their value" [25]. Translation of "Guideline prose into computer executable statements by programmers is complex and arduous because Guideline developers do not plan for algorithmic implementation" [26]. As a result of the poor quality of

Guideline structuring, Clinical Guidelines were often found not to "address all possible situations comprehensively, or provided alternative actions for the same antecedent" [26] when attempting to formalize it.

As a consequence, it has been found essential to verify the internal structure of a Guideline prior to its computerization. One way to verify the internal consistency of a text-based Guideline is to use a Decision Table [27] which makes explicit the different Clinical situations described by the Guideline.

4.1 Brief History of Clinical Guidelines Computerization

The first approach used to encode Clinical Guidelines electronically was developed for the HELP (Health Evaluation through Logical Processing) system at the LDH hospital in Utah [28]. This system was made available to Healthcare practitioners as early as 1975 in order to provide limited decision support at the point of care in the form of notification to clinicians "about events or conditions such as abnormal laboratory results or potentially dangerous drug interactions" [11]. The HELP system used the rules-based "if…then" approach to organize, represent and evaluate the contents of a Clinical Guideline.

At about the same time as the HELP system was developed, another rule-based approach to Clinical Guidelines computerization, the Regenstrief Medical Records System (RMRS) was created. It was developed to optimize information content and delivery to the medical practitioner. Consistent with this goal, the system was created with the ability to represent "current medical knowledge (text book information, published literature) in a codified and active form (*e.g.*, Guidelines) linked to specific patient states with the goal of improving clinical care and assess patient outcome" [11]. Today, this includes "a variety of computer-based informational feedback and interventions designed to change practitioner and/or patient behavior"[2].

4.2 The Arden Syntax

The Arden Syntax is perhaps the best known language for representing clinical knowledge required to create patient-specific DSS. The initial version of the Arden Syntax was based largely on the encoding scheme for generalized decision support used in the HELP system [29]. The Arden Syntax itself was developed in the 1980s and was "accepted as a standard by the ASTM in 1992" and then later by HL7[3]. The perceived modular independence of "if…then" rules led to the "development of the Arden Syntax for encoding MLM as a formalism for sharing medical knowledge used for making decisions" [30]. We may notice that in this case, the target user of the Arden Syntax is the clinician with little or no programming training. This is why it is not a full-feature programming language; for ex-

[2] www.regenstrief.org

[3] http://www.hl7.org

ample, it does not include complex structures to preserve its readability and its proximity to natural language.

4.3 Medical Logic Modules and Clinical Guidelines

The Arden Syntax uses MLM to help providers make medical decisions, by using these models to generate alerts for abnormal laboratory results, drug interactions, diagnostic interpretations, etc. As a standardization tool, the Arden Syntax can be used to relate the contents of Clinical Guidelines to MLM embedding the standardized knowledge they convey. Each MLM is written to behave like a single rule, as instructions within a MLM execute sequentially until a specific outcome is reached.

Each MLM uses four main slots: an "evoke slot", a "data slot", a "logic slot" and an "action slot". The "evoke slot" acts as the trigger, and the "data slot" acts as a retriever of the relevant data from the database. The "logic slot" uses the data in the "if" part of the logic statement and the "action slot" executes the "then" piece of the statement to deliver a reminder or an alert based on the data (*e.g.*, to administer a follow-up preventive exam) or a specific Clinical Guideline recommendation (*e.g.*, dosage of medication). A triggering event can be as simple as the database entry of a diagnosis or another patient data variable (such as a lab test result).

5 Computational Tools Assisting Clinical Guidelines' Structuring

Clinical Guidelines are usually structured to highlight the various decision steps and the sequencing of therapeutic lines. This has induced substantial research in the use of computational tools assisting their structuring. The most comprehensive is the Guideline Elements Model (GEM) [31], which is an XML framework containing more than 100 mark up elements. This model facilitates the encoding of Clinical Guidelines and supports the automatic processing of marked up Clinical documents.

5.1 Guideline Markup Languages

Guideline Markup Languages became available after the widespread availability of HTML / XML browsers. Guideline Markup Languages use a flexible coding system that allows for mark up "tags" to designate concepts and relationships between concepts directly from a text-based Clinical Guideline. Once encoded, the hierarchical structure of markup languages allows the visualization of Clinical Guidelines at "different formats at various levels of detail according to the needs of the practitioner, while preserving their originally published form" [32]. This

constitutes the essence of the mark up approach to Clinical Guideline representation, *i.e.* the ability to impose a structure that defines relationships between the data. With its flexibility and ability to define its own data elements (the definition takes place in a Document Type Definition, or DTD, attached to the mark up document), the markup language approach is designed to allow easy integration into any clinical information system.

5.2 Hypertext Guideline Markup Language

An example of a Guideline Markup Languages is the Hypertext Guideline Markup Language or HGML [32]. The HGML method of Guideline representation is "based on the markup concept for converting text-based Clinical Guidelines into a machine-operable form" by using the markup language ability to allow Guideline authors to mark up "document content and other relevant data (meta-data)" [32]. HGML is eXtensible Markup Language (XML) compatible, which allows for XML libraries and built-in user interface tools within a Web browser to allow the Guideline developer to simply "tag" the Guideline using the HGML structure. HGML uses the standard markup language format. Additional attributes can appear within the begin/end tag structure and are denoted using square brackets when they are optional.

Other advanced HGML tags are available to "facilitate inferences about the Guideline content, identifying levels of evidence for recommendations, and providing links to other documents or even decision theoretic models and simulations" [32].

In summary, HGML allows a textual Guideline to be tagged essentially to produce a structured version of the Clinical Guidelines. Markup language tags also allow representation of Guidelines in several alternative formats at differing levels of detail. The ability to define relationships between tags (or datatypes) is an important differentiation from rule-based languages.

5.3 Guideline Elements Model (GEM)

GEM [31] is another example of a Guideline markup language which uses XML to structure the heterogeneous knowledge contained in Clinical Guidelines. GEM is based on a hierarchy with 9 major branches as shown in Figure 12: *Identity*, *Developer*, *Purpose*, *Intended Audience*, *Method of Development*, *Target Population*, *Testing*, *Review Plan*, and *Knowledge Components*.

The *Knowledge Components* section represents recommendation's logic and constitutes "the essence of practice Guidelines". This section comprises the decision-making branch as it is used to represent the Clinical Guideline's recommendations, definitions, and algorithms. Each of the *Knowledge Components* uses

"elements" (or tags) to describe specific terms that are defined by the National Guidelines Clearinghouse[4], a controlled vocabulary source.

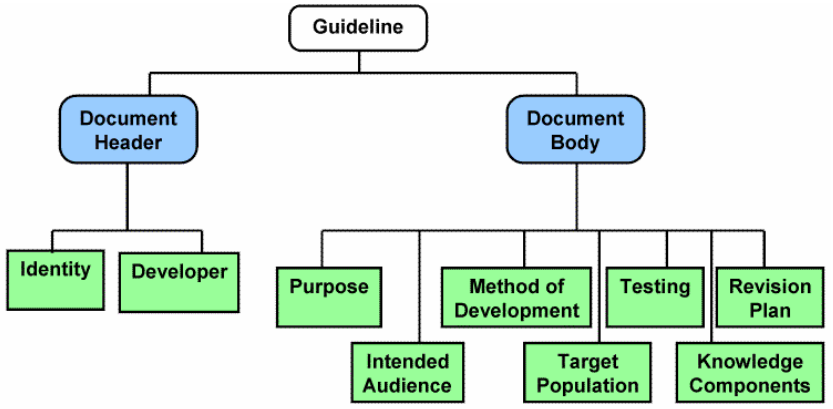

Fig. 12. Representation of the GEM hierarchy.

GEM is intended to facilitate the translation of natural language Guidelines into a standard, computer interpretable, format. One advantage of this format is that it can encode considerable information about Guideline recommendations in addition to the recommendations themselves, including the rationale for each recommendation, the quality of the evidence that supports it, and the recommendation strength assigned to it. GEM encoding of Guideline knowledge is possible through a mark up process that does not require programming skills. In particular, the GEM-Cutter[5] application is an XML editor that facilitates such Guideline mark up.

GEM is also intended for use throughout the entire Guideline life cycle, from Guideline authoring to their dissemination, implementation, and maintenance. In 2002, GEM became an international ASTM standard for the representation of Clinical Guidelines in XML format.

The GEM team is currently investigating reusable methods to facilitate Guideline authoring and implementation using GEM.

However, Guideline marking up has several limitations. Although it has been found comprehensive enough to model the information content of Clinical Guidelines, substantial variation is still observed in the creation of the GEM encoded instance of a given Clinical Guideline by different users [20]. As the model is simply an abstraction of the Guideline document, GEM alone does not support the resolution of ambiguities present in many textual Clinical Guidelines.

[4] http://www.guideline.gov/

[5] http://ycmi.med.yale.edu/GEM/

6 Extending the GEM Model to Generate a Rule Base System

As a preliminary research, we have proposed an extension of the GEM model refining the model's granularity through additional attributes [33]. The aim is to automatically generate a set of canonical decision rules from Clinical Guideline contents: this generation is made possible by the encoding of Guideline contents which can be used to instantiate rules formalisms.

The *Knowledge Components* section is one of the GEM aspects that we extended, focusing on elements signaling therapeutic decisions. We extended the *Conditional* element that represents recommendations applicable under specific circumstances, by adding new attributes to it (Figure 13).

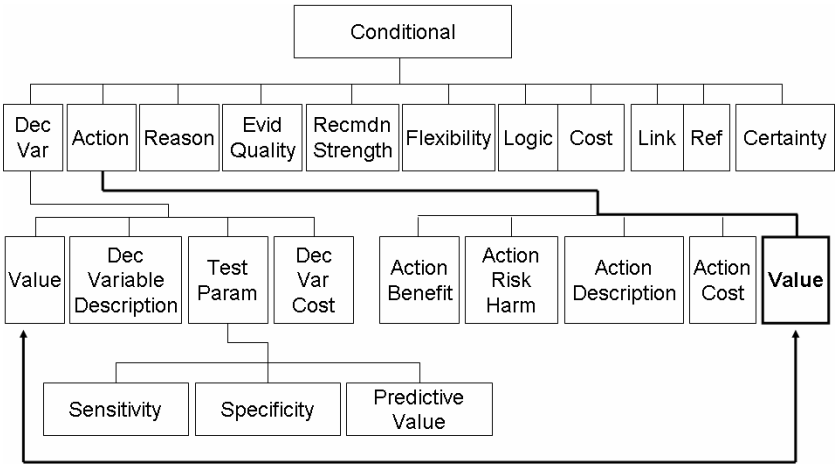

Fig. 13. The extended GEM DTD: the *value* sub-element (for the *action* element) bridges the gap between the decision variable and the action.

Only a few sub-elements are actually used: *decision.variable* (to describe elements of the decision), *action* (to describe the recommended therapy), *recommendation.strength* (to quantify the level of proof). In the GEM DTD, *Conditional* recommendations mainly rely on *decision.variable* and *action* elements. Decision variables are described by a *value*, a *description*, *test parameters* and a *cost*. Actions descriptions are structured through various fields, *i.e. benefit*, *risk, description* and *cost*, which can be grouped together into a single *action parameter* field.

The rationale for the generation of IF-THEN rules is the existence of decision variables in the Clinical Guidelines. More precisely, decision rules are represented as IF-THEN-WITH statements, where the condition (IF) part corresponds to a set of *decision.variable* elements of the GEM DTD, the action (THEN) part corresponds to a set of *action* elements, and the evidence (WITH) part corresponds to the *recommendation.strength* element [33]. To enable the generation of such rules

from a GEM encoded instance, it has been necessary to modify the GEM encoding scheme to reflect the importance of decision variables, and obtain the same structure for both decision variables and actions in the DTD. This is why we first extended the original GEM DTD. We needed a homogeneous data model for *decision.variable* and *action* elements: as the decision variable contains a *value* subelement, we added a similar field to the *action* element.

Even though this extension appears simple in terms of the additional categories introduced, its real power derives from the additional level of structuring which has a strong impact on the elicitation of rule content.

Another extension to the GEM model concerns the structure of actions, represented through the notion of therapeutic strategy (lines of treatment and level of intention). In the example of chronic diseases, therapeutic recommendations depend on the patient state and on her therapeutic history (*e.g.* inadequacy of previous treatment). To resolve Guideline ambiguities in the presentation of the chronological steps of the recommended therapy, we proposed a framework formalizing the therapeutic strategy for a given patient profile. A therapeutic strategy is represented by an ordered sequence of therapeutic lines; each therapeutic line is composed of a set of treatments ordered according to therapeutic levels of intention. Depending on a patient's clinical situation and her response to the ongoing therapy, the recommended treatment may either correspond to the next level of intention within the same therapeutic line or the first level of intention of the following therapeutic line [34].

We started by producing a new encoded instance for the Canadian Clinical Guidelines for hypertension management using our extension of GEM. We then developed a module to automatically derive decision rules from this GEM encoded instance. Finally, as a preliminary form of validation, the rule base automatically generated compared favorably with the manual generation of decision rules [35].

GEM already appears to facilitate the encoding of Clinical Guidelines, which support various aspects of Guideline computerization. Our proposed extension can further support the computerization of Clinical Guidelines through the various aspects of their life cycle, from production to consultation and use.

7 Development of Intelligent Editing Tools for Clinical Guidelines

An important aspect of Clinical Guidelines' computerization consists in assisting expert physicians in the production of Clinical Guidelines and their correct encoding in formats such as GEM. We observed several limitations during the manual encoding of Clinical Guidelines. Tools for facilitating the translation of text into a computable format have been developed but currently rely on a "manual" process as well, simply providing a graphical user interface. In the next section, we introduce a semi-automatic tool improving this process.

7.1 Supporting Knowledge Extraction from Clinical Guidelines

Our idea is to support the automatic process of knowledge acquisition from Clinical Guidelines (Figure 14) which is poised to play an important role in the Clinical Guidelines life cycle.

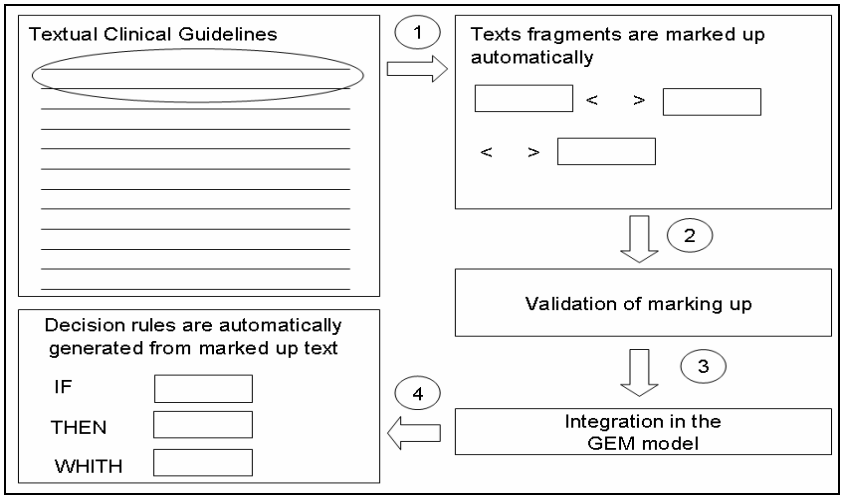

Fig. 14. Knowledge extraction from text: generation of decision rules.

This research may thus enhance existing tools such as GEM-Cutter which is a tool to facilitate the transformation of Clinical Guidelines information into GEM formats. But it still relies on a human step to paste text from Clinical Guidelines (containing appropriate knowledge) into the GEM markers. This is why we proposed to use text processing techniques to identify syntactic structures signaling GEM markers: because of the nature of the Guideline, these are referred to as *deontic operators*. Their automatic identification can support the development of intelligent editing tools for Clinical Guidelines.

7.2 Knowledge Extraction from Texts

Our source of inspiration was research in document processing, essentially for knowledge extraction from prescriptive texts similar to Clinical Guidelines but in other application domains. We based our work on the study of Moulin and Rousseau [36] describing a method to automatically extract knowledge from legal narratives texts based on "*deontic operators*". The following hypotheses have been raised by Moulin and Rousseau: "(i) some prescriptive texts, such as regulation texts, have a form and a content which can be transformed to create at least the kernel of knowledge bases; (ii) the statements of a regulation text can be analyzed

in a systematic way in order to derive knowledge structures which are logically equivalent to the original text and which can be used by an inference engine; (iii) there is a similarity between the way a text is organized and the way a knowledge base can be structured".

In order to verify these hypotheses they developed a knowledge-acquisition system, which transforms a prescriptive text into a knowledge base that can be exploited by an inference engine.

Contents of prescriptive statements are specified by "normative propositions" which in the French texts studied by Moulin and Rousseau manifest themselves through verbs such as "pouvoir" *(be allowed or may)*, "devoir" *(should or ought to)*, "interdire" (*forbid*). The only category of statements whose syntactic form is, according to Kalinowski, typical of norms is deontic propositions. Kalinowski indicates that "whatever kind of norm we consider (moral, juridical, technical or other), the norm may be examined from the point of view of its syntactic structure on one hand, and, on the other hand, from the perspective of inferences in which they are eventually premises or conclusions" [37]. The three most frequent deontic modalities are thus introduced: obligation, interdiction and permission.

Moulin and Rousseau showed that knowledge extraction from legal texts based on deontic operators is a good way to resolve problems of knowledge acquisition from texts, without having to take into account the detailed meaning of recommendations.

7.3 Knowledge Extraction from Clinical Guidelines: a Succession of Steps

The first step consists in analyzing the regularities in Clinical Guidelines, *i.e.* concordances between verbs and expressions. This is done using a Concordance Program. In a second step, using results from the previous analysis, we defined Finite State Transition Networks (FSTN) to represent the surface presentation of deontic operators. We developed a system that runs FSTN to recognize syntactic expressions of deontic operators and automatically marks up Clinical Guidelines in terms of these operators.

7.3.1 Concordance in Clinical Guidelines

As a preliminary study, we used a corpus of 17 French documents such as Clinical Guidelines, consensus conference and medical teaching material (in the field of diabetes, hypertension, asthma, dyslipidemia, epilepsy, renal disease). This enables to collect examples of variability in the authoring of medical documents. We used the "Simple Concordance Program (4.07)[6]" to analyze these documents. This program provides scopes for each word in the corpus.

The aim of this step is also to verify Moulin and Rousseau's hypotheses in the case of Clinical Guidelines. We found several regularities specific to Clinical Guidelines which reproduced the patterns identified by Moulin and Rousseau.

[6] http://asia.cnet.com/downloads/pc/swinfo/0,39000587,20043053s,00.htm

This confirms that we may adapt the deontic logic approach to the processing of textual Clinical Guidelines.

Based also on our previous studies, we focused on verbs that produce decision rules and thus allow a selection of a sub-set of deontic operators. In order to consider deontic operators specific to medicine, we identified the following deontic operators in French: "devoir" (*should or ought to*), "pouvoir" (*be allowed or may)*. However, we noticed that their syntax tends to be specific to the medical context. We observed that a deontic operator is most often followed by the auxiliary "être" (*be*) with a specific verb occurring in its past participle form. We categorized a sub-set of deontic operators associated to previously identified deontic operators. For example, in French "être recommandé" (*be recommended*), or "être prescrit" (*be prescribed*), "être conseillé" *(be advised)*, "être préféré" (*be preferred*). But this syntax is not the only one encountered. We thus collected a complete set of syntactic expressions for deontic operators to built FSTN that will mark up their occurrence in Clinical Guidelines.

7.3.2 FSTN for Marking Up Clinical Guidelines

We decompose the process of marking up in two stages. We first mark up deontic operators, and subsequently mark up their scope in the sentence. A scope that precedes a modal operator is called front-scope, whereas the back-scope corresponds to a scope which follows the operator.

Marking Up Deontic Operators

The first stage applies FSTN to mark up deontic operators in the sentence, as in the example shown in Figure 15.

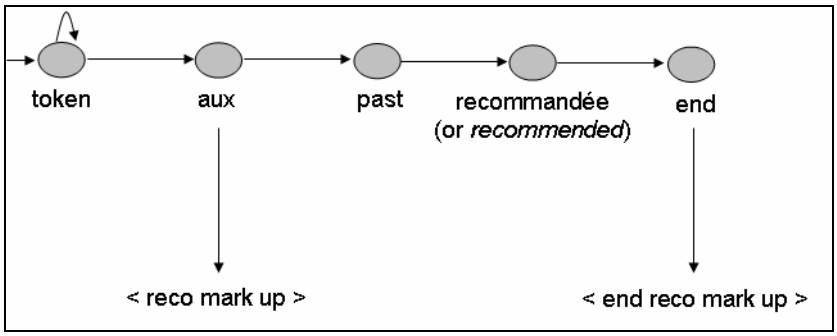

Fig. 15. FSTN for recognizing deontic operator.

We also sub-categorized deontic operators into monadic / dyadic operators, and active / passive forms of the sentence. This sub-categorization assists the marking up of front-scope and back-scope, and the correct identification of the type of information contained in each scope. We defined, as Moulin and Rousseau, an op-

erator scope as the part of the sentence on which the modal operator is applied. Scopes thus correspond to the free text content of deontic expressions.

Marking Up Scopes

The second stage further applies FTSN to mark up scopes in the sentence, as shown in Figure 16, *i.e.* front and back scopes, by the simple processing of marked up operator types.

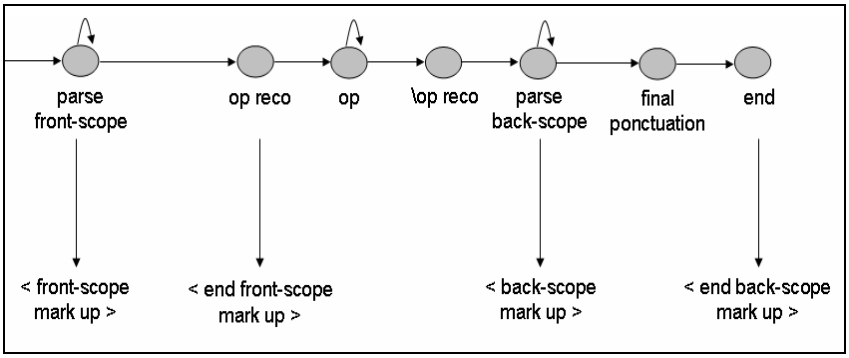

Fig. 16. FSTN for recognizing scopes.

7.3.3 Knowledge Extraction Based on FSTN

The system that we developed, based on FSTN as described above, leads to the automatic marking up of Clinical Guidelines. This system is structured as follows: (i) external files containing FSTN definition and syntactic elements corresponding to their nodes, and text to be tagged; (ii) the FSTN parser, implemented using generic functions.

FSTN Structure

The definition of FSTN is contained in an external file text. We adopt the following formalism: "[" indicates the beginning of the pattern, "]" its end, and "][" words that are ignored during parsing. For example, "est recommandé" (*is recommended*) corresponds to [[aux past]] (*aux* means auxiliary and *past* means past participle form) whereas "est généralement recommandé" (*is generally recommended*) corresponds to [[aux][past]]. This formalism has a simple structure and allows us to represent all patterns identified in the corpus.

Grammar

The grammar is also contained in an external text file. We identified terms, using our previous analysis with the concordance program. As an example, we present below some rewriting rules:

auxiliaire → est | sont	*auxiliairy → is | are*
deontic_operator → peut | peuvent	*deontic_operator → may*
infinitif → constituer | recommander	*infinitive → constitute | recommend*
participe_passé → constitué | recommandé	*past → constituted | recommended*

Internal Structure

We defined functions reading patterns, and rules according to current and previous context. We first generate FSTN from their definition files and then use these functions to parse the document. A function marking up deontic operators is triggered when a pattern corresponding to its FSTN definition has been recognized. Additional functions mark up the front-scope and the back-scope of that sentence.

We also defined rules to identify monadic and dyadic forms of deontic operators. Monadic and dyadic forms are not specific to deontic operators in medical contexts. For example, we defined passive forms whenever the "be" verb occurs after a deontic operator and an auxiliary occurs before the deontic operator. A dyadic form in active voice may be defined by the presence of an expression before and after deontic verbs; it can also be defined whenever expression such as a condition (whether) or an exception (except) occurs before deontic verbs. The monadic form is defined when a pronoun or an expression occurs before a deontic operator for a sentence in the active voice.

7.4 Example

Deontic Operator Considered

The deontic operator "est recommandé" in French (*is recommended*) is first automatically marked up, as in the example below. The process is based on the FSTN described in Figure 15. As an example, in French:

« Dans une quatrième étape du traitement, en cas d'échec de la bithérapie orale maximale, la mise à l'insuline **<Op Reco>** est recommandée **</Op Reco>**, sauf cas particuliers. »

For the fourth step of treatment, if optimal oral bitherapy is inefficient, insulin ***<Op Reco>*** *is recommended* ***</Op Reco>****, except for specific situations.*

Operator Scopes

The second step identifies scopes according to the previous marking up of deontic operators (cf. Figure 16), *i.e.* in this case "Dans une quatrième étape du traitement, en cas d'échec de la bithérapie orale maximale, la mise à l'insuline" (*For the fourth step of treatment, if optimal oral bitherapy is inefficient, insulin*) corresponds to the front-scope and ", sauf cas particuliers" (*, except for specific situations*) constitutes the back-scope. Hence the marked up text:

<Front Scope> Dans une quatrième étape du traitement, en cas d'échec de la bithérapie orale maximale, la mise à l'insuline **</Front Scope> <Op Reco>** est recommandée **</Op Reco> <Back Scope>**, sauf cas particuliers **</Back Scope>**.

<Front Scope> *For the fourth step of treatment, if optimal oral bitherapy is inefficient, insulin* ***</Front Scope> <Op Reco>*** *is recommended* ***</Op Reco> <Back Scope>****, except for specific situations* ***</Back Scope>****.*

7.5 Future Directions

We are currently developing a user-friendly interface to provide a more adequate tool for the interactive marking up of Clinical Guidelines. The development of this tool adopts the same approach as Moulin and Rousseau about ambiguous sentences in the text, *i.e.* where the system alerts the user and asks for confirmation of interpretation.

This automatic marking up can play a role through different steps of computerization of Clinical Guidelines. In a further study, we analyzed treatments according to scopes and deontic operators, and found that marked up treatments facilitate the translation into GEM markers.

One application of the semi-automatic marking up of texts is the generation of decision rules. The generation of decision rules itself can play an important role at various steps of the Clinical Guidelines workflow: DSS can be used to assess the consistency of textual Guidelines at the time of writing. From a different perspective, they can also assist the Guideline user at the point of care. In that sense the case study in automatic rule base generation we have presented is relevant to the overall problem of Guideline computerization.

8 Conclusion

In this chapter, we have reviewed several aspects of the computerization of Clinical Guidelines. As many studies have addressed the computerization of Clinical Guidelines, we tried to account for different standpoints on their computerization and their dissemination. As a synthesis of approaches that we described in this chapter, we may observe that the Arden Syntax and GLIF approaches focus on Clinical Guidelines standardization, and PRO*forma* and GUIDE on execution aspects. However, the representation model of the Arden Syntax differs from any other approach, as it is the only approach that models each Guideline as an independent modular rule. As a result, the Arden Syntax is most suitable for representing simple Guidelines such as alerts in reminder systems.

The GLIF, PRO*forma*, and GUIDE approaches all model Clinical Guidelines in a similar way, in terms of primitives (steps, tasks) that describe the control structure of a Guideline. GUIDE represents Clinical Guidelines through Petri nets integrated into a workflow; PRO*forma* relies on a task model to structure the Guide-

lines contents; GLIF defines different layers of abstraction, but uses flowcharts as a procedural representation of Guidelines contents.

All approaches have a formal syntax for their representation language: PRO*forma* uses BNF, GLIF uses UML, and GUIDE uses workflow process definition language.

However, we observed that the tools developed to assist the translation of Clinical Guidelines into computable format only operate manually. This is why our inspiration was driven from research in document processing, in particular techniques for information extraction from texts. In the FASTUS system [38], Hobbs et al. explain that in information extraction "generally only a fraction of the text is relevant, information is mapped into a predefined, relatively simple, rigid target representation; this condition holds whenever entry of information into a database is the task; the subtle nuances of meaning and the writer's goals in writing the text are of at best secondary interest". We thus adopted an approach based on surface structure, and on the study of Moulin and Rousseau, who have a compatible approach due to the fact that they have worked on prescriptive texts.

We claim this approach can improve existing tools such as GEM-Cutter. Clinical Guidelines show regularities in their authoring, and this is the reason why we follow Moulin and Rousseau in their use of deontic logic. Using our system, we are able to automatically mark up relevant information of Clinical Guidelines and translate automatically to a GEM encoding.

The computerization of Clinical Guidelines should address their entire workflow, from their production and encoding in document exchange formats, to their on-line consultation and their use for knowledge elicitation in DSS. Further work should also investigate the relations between their workflow and Health Information Systems.

References

1. Keckley PH. Evidence-Based Medicine in Managed Care: a Survey of Current and Emerging Strategies. Medscape General Medicine 6 (2004) 56.
2. McGlynn EA, Asch SM, Adams J, Keesey J, Hicks J, DeCristofaro A, and Kerr EA. The Quality of Health Care Delivered to Adults in the United States. The New England Journal of Medicine 348 (2003) 2635-2645.
3. Midwest Business Group on Health. Reducing the Costs of Poor-Quality Health Care Through Responsible Purchasing Leadership. Chicago, Il: Midwest Business Group on Health (2003).
4. Sackett DL, Straus SE, Richardson WS, Rosenberg W, and Haynes RB. Evidence-Based Medicine: How to Practice and Teach EBM. Edinburgh: Churchill Livingston (2000).
5. Grimshaw JM and Russel IT. Effect of Clinical Guidelines on Medical Practice: a Systematic Review of Rigorous Evaluations. Lancet 342 (1992) 1317-1322.
6. Cabana MD, Rand CS, Powe NR, Wu AW, Wilson MH, Abboud PAC, and Rubin HR. Why Don't Physicians Follow Clinical Practice Guidelines? A Framework for Improvement. The Journal of the American Medical Association 15 (1999) 1458-1465.

7. Medin DL and Ross BH. The Specific Character of Abstract Thought: Categorization, Problem Solving, and Induction. In: R.J. Sternberg, Ed. Advances in the Psychology of Human Intelligence. Hillsdale, NJ: Lawrence Erlbaum (1989) 189-223.
8. Patel VL, Arocha JF, Diermeier M, Greenes RA, and Shortliffe EH. Methods of Cognitive Analysis to Support the Design and Evaluation of Biomedical Systems: the Case of Clinical Practice Guidelines. Journal of Biomedical Informatics 34 (2001) 52-66.
9. Bennett JW and Glasziou PP. Computerised Reminders and Feedback in Medication Management: a Systematic Review of Randomised Controlled Trials. The Medical Journal of Australia 178 (2003) 217-222.
10. Lobach DF and Hammond WE. Computerized Decision Support Based on a Clinical Practice Guideline Improves Compliance with Care Standards. The American Journal of Medicine 102 (1997) 89-98.
11. Elkin PL, Peleg M, Lacson R, Bernstam E, Tu S, Boxwala A, Greenes R, and Shortliffe EH. Toward Standardization of Electronic Guidelines. MD Computing 17 (2000) 39-44.
12. Shiffman RN, Shekelle P, Overhage JM, Slutsky J, Grimshaw J, and Deshpande AM. Standardized Reporting of Clinical Practice Guidelines: A Proposal from the Conference on Guideline Standardization. Ann Intern Med 139 (2003) 493-498.
13. Tierney WM, Overhage JM, Takesue BY, Harris LE, Murray MD, Varco DL, and McDonald CJ. Computerizing Guidelines to Improve Care and Patient Outcomes: The Example of Heart Failure. Journal of the American Medical Informatics Association 2 (1995) 316-322.
14. Patel VL, Arocha JF, Diermeier M, How J, and Mottur-Pilson C. Cognitive Psychological Studies of Representation and Use of Clinical Practice Guidelines. International Journal of Medical Informatics 63 (2001) 147-167.
15. Ohno-Machado L, Gennari JH, Murphy SN, Jain NL, Tu SW, Oliver DE, Pattison-Gordon E, Greenes RA, Shortliffe EH, and Barnett GO. The Guideline Interchange Format: A Model for Representing Guidelines. Journal of the American Medical Informatics Association 5 (1998) 357-372.
16. Fox J, Johns N, and Rahmanzadeh A. Disseminating Medical Knowledge: the PROforma Approach. Artificial Intelligence in Medicine 14 (1998) 157-81.
17. Quaglini S, Stefanelli M, Cavanelli A, Micieli G, Fassino C, and Mossa C. Guideline-Based Careflow Systems. Artificial Intelligence in Medicine 20 (2000) 5-22.
18. Hripcsak G, Ludemann P, Pryor TA, Wigertz OB, and Clayton PD. Rationale for the Arden Syntax. Computers and Biomedical Research 27 (1994) 291-324.
19. Peleg M, Tu S, Bury J, Ciccarese P, Fox J, Greenes RA, Hall R, Johnson PD, Jones N, Kumar A, Miksch S, Quaglini S, Seyfang A, Shortliffe EH, and Stefanelli M. Comparing Computer-Interpretable Guideline Models: A Case-Study Approach. Journal of the American Medical Informatics Association 10 (2003) 52-68.
20. Karras BT, Nath SD, and Shiffman RN. A Preliminary Evaluation of Guideline Content Mark-Up Using GEM - An XML Guideline Elements Model. Proceedings of the American Medical Informatics Association (2000) 413-417.
21. Peleg M and Kantor R. Approaches for Guideline Versioning Using GLIF. Proceedings of the American Medical Informatics Association (2003) 509-513.
22. Peleg M, Boxwala AA, Ogunyemi O, Zeng Q, Tu S, Lacson R, Bernstam E, Ash N, Mork P, Ohno-Machado L, Shortliffe EH, and Greenes RA. GLIF3: The Evolution of a Guideline Representation Format. Proceedings of the American Medical Informatics Association (2000) 645-649.

23. Fox J, Beveridge M, and Glasspool D. Understanding Intelligent Agents: Analysis and Synthesis. Ai Communications 16 (2003) 139-152.
24. Ciccarese P, Caffi E, Boiocchi L, Quaglini S, and Stefanelli M. A Guideline Management System. In: M. Fieschi et al., Eds. IOS Press. Proceedings of International Medical Informatics Association (2004) 28-32.
25. Shiffman RN and Greenes RA. Improving Clinical Guidelines with Logic and Decision-Table Techniques: Application to Hepatitis Immunization Recommendations. Medical Decision Making 14 (1994) 245-254.
26. Shiffman RN, Brandt CA, Liaw Y, and Corb GJ. A Design Model for Computer-based Guideline Implementation Based on Information Management Services. Journal of the American Medical Informatics Association 6 (1999) 99-103.
27. Shiffman RN. Representation of Clinical Practice Guidelines in Conventional and Augmented Decision Tables. Journal of the American Medical Informatics Association 4 (1997) 382-393.
28. Kuperman GJ, Gardner RM, and Pryor TA. The HELP System. Springer-Verlag, New York (1991).
29. Gardner RM, Pryor TA, and Warner HR. The HELP Hospital Information system: update 1998. International Journal of Medical Informatics 54 (1999) 169-182.
30. Pryor TA and Hripcsak G. The Arden Syntax for Medical Logic Modules. International Journal of Clinical Monitoring and Computing 10 (1993) 215-224.
31. Shiffman RN, Karras BT, Agrawal A, Chen R, Marenco L, and Nath S. GEM: A Proposal for a More Comprehensive Guideline Document Model Using XML. Journal of the American Medical Informatics Association 7 (2000) 488-98.
32. Hagerty CG, Pickens D, Kulikowski C, and Sonnenberg F. HGML: A Hypertext Guideline Markup Language. Proceedings of the American Medical Informatics Association (2000) 325-329.
33. Georg G, Séroussi B, and Bouaud J. Extending the GEM Model to Support Knowledge Extraction from Textual Guidelines. International Journal of Medical Informatics (2004) In Press.
34. Georg G, Séroussi B, and Bouaud J. Interpretative Framework of Chronic Disease Management to Guide Textual Guideline GEM-Encoding. Studies in Health Technology and Informatics 95 (2003) 531-536.
35. Georg G, Séroussi B, and Bouaud J. Does GEM-Encoding Clinical Practice Guidelines Improve the Quality of Knowledge Bases? A Study with the Rule-Based Formalism. Proceedings of the American Medical Informatics Association (2003) 254-258.
36. Moulin B and Rousseau D. Knowledge Acquisition from Prescriptive Texts. Proceedings of the Association for Computing Machinery (1990) 1112-1121.
37. Kalinowski G. La Logique Déductive. Presses Universitaires de France (1996).
38. Hobbs JR, Jerry R, Appelt DE, Bear J, Israel D, Kameyama M, Stickel M, and Tyson M. FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text. In E. Roche and Y. Schabes, Eds. Finite State Devices for Natural Language Processing, MIT Press, Cambridge, Massachusetts (1996) 383-406.