

An Active Database Approach to Computerised Clinical Guideline Management

Kudakwashe Dube and Bing Wu

Department of Computer Science, School of Computing, Dublin Institute of Technology,
Kevin Street, Dublin 8, Ireland.

Abstract. This paper presents a generic approach, and a case study practising the approach, based on a unified framework harnessing the event-condition-action (ECA) rule paradigm and the *active database* for the management of computer-based *clinical practice guidelines* and protocols (CPGs). The CPG management is cast into three perspectives: *specification*, *execution* and *manipulation*, making up three management planes of our framework. The ECA rule paradigm is the core CPG representational formalism while the active database serves as the kernel within the CPG management environment facilitating integration with electronic healthcare records and clinical workflow. The benefits of the approach are: *flexibility* of CPG management; *integration* of CPGs with electronic patient records and clinical workflows; and *incorporation* of CPG management system into computerised healthcare systems.

1 Introduction

Clinical practice guidelines (CPGs) encapsulate medical knowledge about when clinicians and patient should carry out specific activities in disease management. CPGs are effectively “knowledge models of preferred processes of care” [1], which can guide patient care activities and clinical workflow or care flow [2]. In practice, a CPG is “a method that identifies actions that are to be performed and that specifies conditions that govern when it is appropriate to perform them” [3]. Thus, a CPG embodies the core compositional primitives of the *event-condition-action (ECA)* rule paradigm. ECA rules are specified by an event, a condition and an action with the semantic that the event must occur for the action to be executed subject to the condition being satisfied [4]. An *active database management system (ADBMS)* is a DBMS that incorporates the ECA rule mechanism and provides functionalities as stipulated in the Active Database System Manifesto [5]. The ECA rule paradigm could be a strong basis for formalising CPGs while an active database system could be a promising management environment for computerised CPG. The authors’ hypothesise that, in combination, these two could facilitate a solution to the problem of the integration of computer-based CPGs with the electronic patient record and computerised clinical workflow. This paper presents a computerised CPG management approach and a case study which uses a generic, unified management framework and harnesses the ECA rule paradigm and the active database in the CPG management task. The rest of this paper is organised as follows: Section 2 presents a

review of the ECA rule paradigm in computer-based CPG management support; Section 3 presents the generic framework for CPG management; Section 4 presents the method developed in this work for managing the CPGs using the ECA rule paradigm and active databases within the SpEM framework; Section 5 presents a case study in which the proof-of-concepts system is developed and used to manage the microalbuminuria CPG for diabetes patients; and Section 7 summarises and concludes this paper.

2 Relevant Work in Supporting Clinical Guideline Management

This section reviews relevant work in supporting computerised CPG management. Research in computer-based support for the management of CPGs has been on-going for over a decade. A number of approaches and systems have emerged during this period. For a more general review, readers are referred to de Clercq *et al* [6]. Most approaches focus on the formal representation and execution of computerised CPGs. Among them, the Arden Syntax [7], an established HL7 standard [8], and HyperCare [9] are most relevant to this work. While most approaches use control structure primitives [6] to represent guidelines, the Arden Syntax and HyperCare use the ECA rule paradigm. However, the Arden Syntax focuses on CPG specification only and HyperCare focuses on execution only, using an active database system in its implementation. Both the Arden Syntax and HyperCare do not create patient-specific CPG instances. Instead, all rules operate with a global scope covering all patients. Furthermore, these two approaches offer no support for the management of information on both CPG specifications and their instances. Hence, they do not fully exploit all the advantages of the database approach to realise CPG information management. Departing from others, authors of this paper have developed a unified CPG management framework which places equal emphasis on the CPG management perspectives of specification, execution and manipulation and harnesses the active database for supporting all aspects of this framework. In this framework, generic CPG specifications are created and held in the database together with execution state and effects for easy management. The CPG execution aspect supports patient-specific CPG instances and harnesses the active mechanism of the DBMS. Furthermore, database manipulation and query features are exploited for the computerised CPG information management.

3 A Generic Framework for Clinical Guideline Management

This section presents a generic framework, SpEM [10] (Specification, Execution and Manipulation) developed by the authors for supporting the computerised CPG management. As illustrated in Figure 1, the SpEM framework contains 3 planes: *specification* plane; *execution* plane and; *manipulation* plane with the active database as the integrating factor among the three planes.

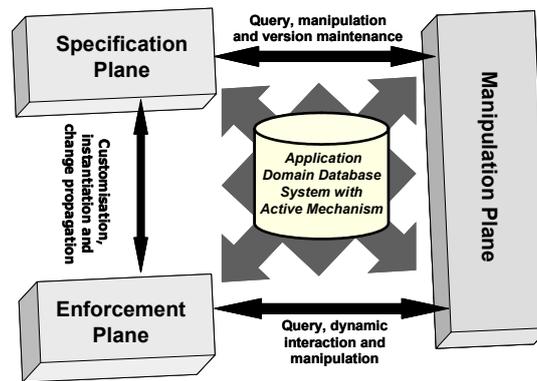


Fig. 1. The SpEM framework.

The CPG management process is fitted into the three planes of the framework. In the *specification plane*, CPGs are translated into formal specifications and held permanently in the database in a manageable form. In the *execution plane*, the stored CPG specifications are used to create patient-specific CPG instances that are executable by an engine based on the active database. In the *manipulation plane*, the specifications and the executing CPG instances are manipulated using supported operations and queries based on the SQL features of the underlying database system. The active database is a base for an execution engine for CPGs. It provides facilities for querying and manipulating information. It can also permanently hold the electronic patient record in addition to CPG information and running processes. Furthermore, it can guarantee future sharing of information through the generic nature of databases and the standard language, the SQL. Table 1. summarises the support for the SpEM framework in major existing CPG approaches and systems. However, these approaches only provide guideline support mainly in terms of the specification and execution of CPGs. Not much attention seems to have been paid to the support for the manipulation and querying of the CPG information and to harnessing of database features for CPG management. Manipulation of the static and dynamic aspects of CPGs is important to allow flexibility and ease-of-use [19], which are some of the major determining factors in the acceptability of guideline systems by clinicians [20].

Table 1. Support for the SpEM framework in existing CPG management systems.

Guideline System	SpEM Framework Support				Use of Database Features
	Key: √ - full support, * - partially supported, X - no support.				
	Specification	Execution	Manipulation		
			Operation	Query	
DILEMMA/PRESTIGE [11]	√	√	X	X	X
EON/Dharma [12]	√	√	X	X	X
PROforma [13]	√	√	X	X	X
GLIF [14]	√	*	X	X	X
Asgaard/ Asbru [15]	√	√	*	*	X
GUIDE [2]	√	√	X	X	X
PROGIDY [16]	√	√	X	X	X
Arden Syntax [7]	√	X	X	X	X
HyperCare [9]	X	√	X	X	*
GASTON [17]	√	√	*	X	X
GLARE [18]	√	√	X	X	X

4 A Method for Computerised CPG Management

This section presents the method for managing computerised CPGs through the SpEM framework. Fig. 2 illustrates how the method permeates the three planes of the framework and allows CPGs to be formally specified, stored, executed with respect to a given patient, and manipulated through querying and operations. Several aspects need to be incorporated as components of the management process for supporting CPG management. Domain knowledge, i.e., expertise and literature on recent advances in medical knowledge, is the main source of CPGs. Its translation into CPGs is done by clinicians and is outside the scope of SpEM. Formal representation and creation of CPG specifications and their storage are important aspects of the computerisation of CPGs. Customisation and execution of computerised CPGs with respect to specific individual patient are vital aspects of the management of computerised CPGs. The manipulation of both the formal specifications and the running process consists of the two aspects: querying; and performing manipulation operations on the CPGs. The process illustrated in Fig. 2 is to ensure that most aspects of the computerised CPGs can be managed. Fig. 2 also illustrates the main functionality for supporting CPG management. The main requirement of the *specification plane* is a declarative language, the clinical Protocol Language, PLAN, together with its model [21]. PLAN was designed to use the ECA rule paradigm as the core representation construct for specifying CPGs. Fig. 3. illustrates the general syntax of PLAN in the Backus-Nuar Form (BNF). A PLAN specification consists of a descriptive header; a set of schedules; and the protocol rule set. Just like a protocol, a schedule is named and consists of an entry criteria and a list of rules each of which is either a dynamic or static rule. For a more detailed discussion of PLAN, readers are referred to Wu and Dube [21]. In the *execution plane*, the relevant PLAN specification is customised and installed for each patient as an instance that runs within the ECA rule mechanism of a database system. The execution of the guideline instance follows the ECA mechanism. Within the *manipulation plane*, provision is made to perform operations and to issue queries using a query and manipulation language that is based on the SQL.

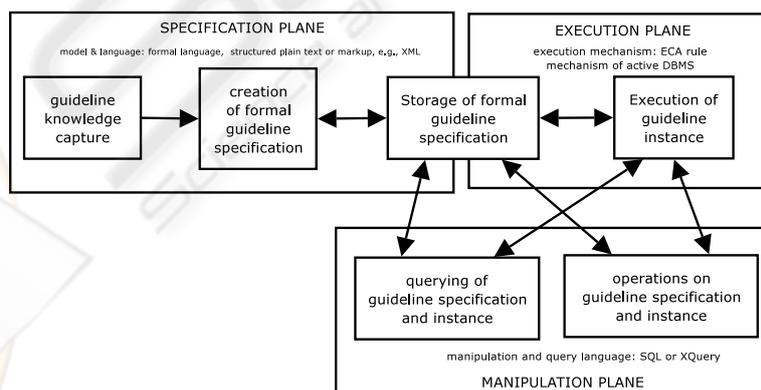


Fig. 2 The management support process in SpEM.

```

<static_rule> ::= ECA rule with time events
<protocol> ::= PROTOCOL <protocol_body> END PROTOCOL.
<protocol_body> ::= <protocol_header> SCHEDULE_SET <schedule_list> END
SCHEDULE_SET <protocol_rule_set>
<protocol_header> ::= <name>; <description>; <creator_detail>; <category>;
<schedule_list> ::= <schedule> | <schedule>; <schedule_list>
<schedule> ::= SCHEDULE <schedule-header>; <entry_criteria>; <rule_list> END SCHEDULE
<protocol_rule_set> ::= RULE_SET <rule_list> END RULE_SET;
<rule_list> ::= <rule> | <rule>; <rule_list>
<rule> ::= <dynamic_rule> | <static_rule>
<dynamic_rule> ::= ECA rule

```

Fig. 3. The high-level BNF syntax of PLAN.

5 Case Study: TOPS and the Management of the Microalbuminuria CPG for Diabetes Patients

This section outlines the design of the proof-of-concepts system, TOPS, and a case study in which TOPS is used to manage the CPG for microalbuminuria in diabetes patients. Fig. 4 illustrates the generic CPG management method that is used in TOPS.

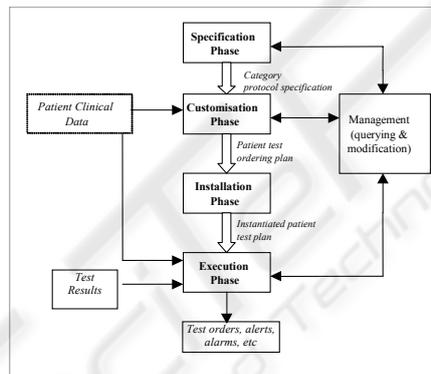


Fig. 4. The CPG Management Process for clinical protocols.

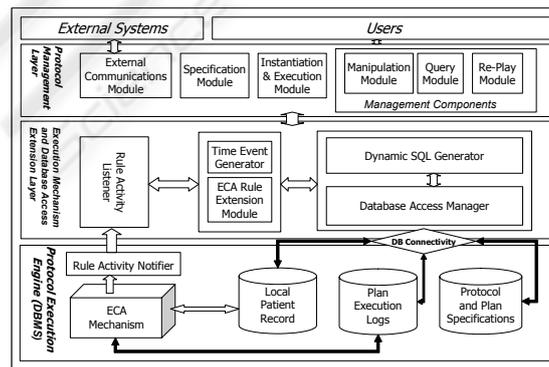


Fig. 5. Architecture of the proof-of-concepts system, TOPS.

In the *specification phase*, CPGs are formally represented and specified. Resulting specification are stored in a database. The domain expert needs to be involved in this process. In the *customisation phase*, the computerised CPG is customised to suit the specific clinical problem suffered by the patient. This phase produces patient-specific CPG specification instances as illustrated in Fig. 4. In the *installation phase*, all the ECA rules in the CPG are added to the active database resulting in an active CPG instance. In the *execution phase*, the execution process proceeds in accordance with the ECA mechanism. The *manipulation phase* includes querying and manipulation operations and permeates all the above phases. The architecture of TOPS has three layers as illustrated in

Fig. 5. External to TOPS, are users and external systems. The top layer is the clinical protocol management functionality for specifying, storing, executing, manipulating and querying CPGs. The middle layer provides services that extend the ECA rule execution mechanism of the underlying DBMS and handles database connectivity. The bottom layer is the CPG execution engine as well as the system database, which is currently based on the Oracle9i DBMS. The architecture provides support for the three planes SpEM. Issues of concurrency and efficiency in rule execution are handled by the Oracle9i DBMS with the exception of ECA rule extensions that are implemented externally.

The SpEM framework in TOPS has been demonstrated through a case study involving the CPG for the diagnosis and management of microalbuminuria (MA) in diabetes mellitus as interpreted by a practising clinician at the local Diabetes Day Clinic in Dublin. Fig. 6. illustrates the state chart for the *microalbuminuria protocol* (MAP). For each state and its associated transitions, rules are designed to perform what must be done when the patient enters, stays and exits from the state. Other rules serve to trigger patient state transitions. Thus, the state chart guides creation of MAP specification. Fig. 7. illustrates the outline structure and content of the resulting MAP specification. The MAP specification used to create a patient plan when a user requests the creation of a MAP instance for a specified patient. Patient plan execution proceeds in an event-driven manner according to the set of ECA rules making up the MAP instance. In the query illustrated in Fig. 8., a patient-specific MAP instance's snapshot at a given time or interval is retrieved. The MAP instance snapshot refers to

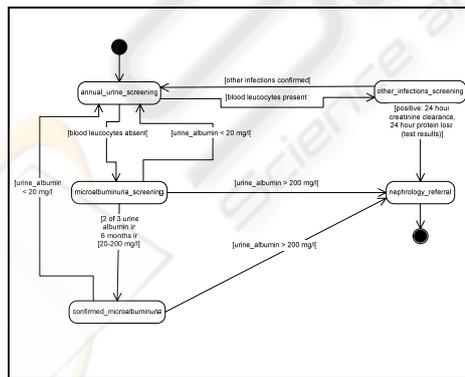


Fig. 6. State chart for the microalbuminuria protocol.

```
@PROTOCOL@ MAP;
DESCRIPTION: This is a protocol for the diagnosis and management of microalbuminuria in
diabetes patients;
CREATOR: DR JOHN NOLAN;
CATEGORY: DIABETIC_NEPHROPATHY;
#SCHEDULE_SET#
^SCHEDULE^ AUS, <AUS_rules> ^END SCHEDULE^
^SCHEDULE^ OIS, <OIS_rules> ^END SCHEDULE^
^SCHEDULE^ MAS, <MAS_rules> ^END SCHEDULE^
^SCHEDULE^ CMA, <CMA_rules> ^END SCHEDULE^
^SCHEDULE^ NPH, <NPH_rules> ^END SCHEDULE^
#END SCHEDULE_SET#
~RULE_SET~
STATIC_RULE AUS1, <body of rule AUS1>;
RULE OIS1, <body of rule OIS1>;
RULE MAS1, <body of rule MAS1>;
RULE CMA1, <body of rule CMA1>;
RULE CMA2, <body of rule CMA2>;
RULE CMA3, <body of rule CMA3>;
RULE CMA4, <body of rule CMA4>;
RULE NPH1, <body of rule NPH1>;
#END RULE_SET#
@END PROTOCOL@

Abbreviations:
AUS - Annual Urine Screening, OIS - Other-Infections Screening, MAS - MicroAlbuminuria
Screening, CMA - Confirmed MicroAlbuminuria, NPH - Nephrology referral
```

Fig. 7. Structure of the Microalbuminuria Protocol in PLAN.

```

TOPS:> query
QUERY:> -->SELECT PLAN FROM SNAPSHOT WHERE TARGET:2005-7-19
01:55:02,2005-7-19 01:55:58; SOURCE:PATIENT_ID=25
processing query ...
PLAN [ PLS25S1$ ] SNAPSHOT @[2005-07-19 22:30:26.91]
rule 1]-->[ 72, PLS25S1$SAUS2, DYNAMIC, READY ]
rule 2]-->[ 73, PLS25S1$SAUS3, DYNAMIC, READY ]
rule 3]-->[ 74, PLS25S1$OIS2, DYNAMIC, READY ]
rule 4]-->[ 75, PLS25S1$OIS3, DYNAMIC, READY ]
rule 5]-->[ 76, PLS25S1$OIS4, DYNAMIC, READY ]
...
[rule 19]-->[ 90, PLS25S1$CMA4a, DYNAMIC, READY ]
rule 20]-->[ 91, PLS25S1$CMA4b, DYNAMIC, READY ]
rule 21]-->[ 92, PLS25S1$SNPH1, DYNAMIC, READY ]
rule 22]-->[ 93, PLS25S1$mainSAUS1, STATIC, EXECUTING ]
END SNAPSHOT FOR PLAN PLS25S1$.
QUERY:> -->exit
TOPS:>

```

Fig. 8. A query for a snapshot of the composition of a CPG instance in TOPS.

```

TOPS:> query
QUERY:> -->SELECT ORDER FOR PATIENT WHERE TARGET:2005-7-16 17:48:30,2005-
7-16 17:51:25; SOURCE:PATIENT_ID=61
processing query ...
launching specialised query handler...
processing ORDER query [ 2005-7-16 17:48:30,2005-7-16 17:51:25 ] for [ PATIENT ]
...
Tests ordered for [PATIENT_ID=61] during time interval [2005-7-16 17:48:30,2005-7-16
17:51:25]
Dip_stick_urine Profile, DSU, 2005-07-16 17:49:28.0
Urinary_Tract_Infection Profile, UTI, 2005-07-16 17:50:06.0
Urinary_Tract_Infection Profile, UTI, 2005-07-16 17:50:06.0
-----
End test listing.
QUERY:> -->EXIT
TOPS:>

```

Fig. 9. A query to find out what clinical test orders where made during the specified time interval.

the instance's current state in terms of rule composition and the status of its rules at that time or interval. The query illustrated in Fig. 9, provides information on what tests were ordered with respect to the specified patient during the given time interval. The term *order* in the query can be generalised to *rule-action* so that one can obtain information on rule actions that have been performed during the specified time interval. In this case study, the use of the highly intuitive UML state chart brings easy communication with domain experts during CPG knowledge elicitation, capture and specification. Subsequent extraction of the relevant ECA rules is also made easier since the state chart naturally supports the ECA rule paradigm [22] and is also easily understood by domain experts. The database offered a uniform and flexible way to access, manipulate and query all information from specification, to executing process state, to data in the patient record. The generation of SQL trigger code implementing ECA rules of the MAP was automatically supported by TOPS and required minimal intervention. This makes it easy for application domain experts to use TOPS with no knowledge of the SQL trigger specification language. However, domain experts still needed to be familiar with the specification language, PLAN, which is closer to their domain language than the SQL. The execution of the rule actions was subject to the availability of the appropriate software module that implements the action. Thus, rule actions in the microalbuminuria CPG needed to be predefined and any new action requires that the module to implement such an action be developed. However, rule actions were designed to be generic and re-usable by other rules in other CPGs. Using the database permits operations and queries on various aspects of the MAP through an SQL-based manipulation language. It was shown that the MAP can be modeled and specified by using the ECA rule paradigm guided by the state chart. This case study demonstrated the applicability of the SpEM framework and the active database in enabling the support for the management of the MAP for diabetes patients.

6 Summary and Conclusion

This paper has presented a unified CPG management framework, SpEM, for computerized CPG management. The paper also presented a generic method with a case study to harness the ECA rule paradigm and active databases to provide computerized CPG support, by following the SpEM. Active databases combine the ECA rule paradigm with data management to present a promising environment for

supporting CGPs and their integration with the electronic medical record and clinical workflow. This work contributes a generic approach with a framework to unify the three CPG management dimensions and an active database method for computational support. The benefits of the approach are: *flexibility* based on CPG information management; *ease of integration* of CPGs with electronic patient records and clinical workflows due to the active database approach; and *ease of incorporation* of CPG management system into the healthcare systems due to ubiquity of database systems within most institutions. Our future work will focus on: 1) finding ways to deploy and evaluate the SpEM and TOPS in a real patient care setting without infringing on confidentiality and proprietary license restrictions on APIs and schemas in existing hospital systems; 2) improving our framework though enriched specification model and language; 3) more efficient methods of exploiting and enhancing the active mechanism by hybridising it with other paradigms; and 4) useful concepts and methods for information manipulation, query and replay.

Acknowledgements

The Arnold F Graves Trustees and the Dublin Institute of Technology are gratefully acknowledged for sponsoring this research work.

References

1. OpenClinical, The Medical Knowledge Crisis and its solution Through Knowledge Management. 2001, OpenClinical.org.
2. Quaglino, S., Stefanelli, M., Cavallini, A., Micieli, G., Fassino, C., and Mossa, C., Guideline-based careflow systems. *Artif Intell Med*, 2000. 20(1): p. 5-22.
3. Pattison-Gordon, E., Cimino, J.J., Hripcsak, G., Tu, S.W., Gennari, J.H., Jain, N.L., and Greenes, R.A., Requirements of a sharable guideline representation for computer applications. 1996, Stanford University: Stanford, California, USA.
4. Widom, J. and Ceri, S., eds. *Active Database Systems: Triggers and Rules for Advanced Database Processing*. 1996, Morgan Kaufmann: San Francisco, USA.
5. Dittrich, K.R., Gatzju, S., and Geppert, A. The Active Database Management System Manifesto: A Rulebase of ADBMS Features. in *2nd Workshop on Rules in Databases (RIDS)*. 1995. Athens, Greece: Springer.
6. de Clercq, P.A., Blom, J.A., Korsten, H.H., and Hasman, A., Approaches for creating computer-interpretable guidelines that facilitate decision support. *May. Artif Intell Med.*, 2004. 31(1): p. 1-27.
7. Clayton, P.D., Pryor, T.A., Wgertz, O.B., and Hripcsak, G. Issues and Structures for Sharing Medical Knowledge Among Decision-making Systems: The 1989 Arden Homestead Retreat. in *Annu Symp Comput Appl Med Care*. 1989.
8. HL7, Arden Syntax for Medical Logic Modules., in *Standards of the Health Level 7*. 1999: USA.
9. Caironi, P.V.C., Portoni, L., Combi, C., Pinciroli, F., and Ceri, S. HyperCare: a Prototype of an Active Database for Compliance with Essential Hypertension Therapy Guidelines. in *AMIA Ann Fall Symposium*. 1997. Philadelphia, PA: Hanley and Belfus.

10. Dube, K., A Generic Approach to Supporting the Management of Computerised Clinical Guidelines and Protocols, PhD Thesis, in Computer Science. 2004, Dublin Institute of Technology: Dublin. p. 324.
11. Gordon, C., Herbert, I., and Johnson, P. Knowledge Representation and Clinical Practice Guidelines: the DILEMMA and PRESTIGE projects. in Medical Informatics Europe. 1996. Copenhagen: IOS Press.
12. Musen, M.A., Tu, S.W., Das, A.K., and Shahar, Y., EON: A component-based approach to automation of protocol-directed therapy. JAMIA, 1996. 3(6): p. 367-88.
13. Fox, J., Johns, N., Rahmzadeh, A., and Thomson, R. PROforma: a method and language for specifying clinical guidelines and protocols. in Medical Informatics Europe. 1996. Amsterdam.
14. Ohno-Machado, L., Gennari, J.H., Murphy, S., Jain, N.H., Tu, S.W., Oliver, D.E., Pattison-Gordon, E., Greenes, R.A., Shortliffe, E.H., and Barnett, G.O., The GuideLine Interchange Format: A Model for Representing Guidelines. JAMIA, 1998. 5(4): p. 357-372.
15. Shahar, Y., Miksch, S., and Johnson, P., The Asgaard Project: A task-specific framework for the application and critiquing of time-oriented clinical guidelines. Artificial Intelligence in Medicine, 1998. 14: p. 29-51.
16. Johnson, P.D., Tu, S.W., Booth, N., Sugden, B., and Purves, I.N., A guideline model for chronic disease management. 1999, Stanford Medical Informatics: Stanford.
17. de Clercq, P.A., Hasman, A., Blom, J.A., and Korsten, H.H.M., Design and implementation of a framework to support the development of clinical guidelines. Int J Med Inf., 2001. 64(2-3): p. 285-318.
18. Terenziani, P., Molino, G., and Torchio, M., A modular approach for representing and executing clinical guidelines. Artif Intell Med., 2001. 23(3): p. 249-76.
19. Quaglioni, S., Stefanelli, M., Lanzola, G., Caporusso, V., and Panzarasa, S., Flexible guideline-based patient careflow systems. Artif Intell Med, 2001. 22(1): p. 65-80.
20. Musen, M.A. Two decades of Models and Components: Why has Automation of Guideline-Directed Care been so Difficult? in 1st European Workshop on Clinical Practice Guidelines and Protocols (EWGLP 2000). 2000. Leipzig, Germany: IOS Press.
21. Wu, B. and Dube, K. PLAN: a Framework and Specification Language with an Event-Condition-Action (ECA) Mechanism for Clinical Test Request Protocols. in 34th Hawaii International Conference on System Sciences (HICSS-34): the Mini-Track in Information Technology in Healthcare. 2001. Maui, Hawaii: IEEE Computer Society, Los Alamitos, California.
22. Berndtsson, M. and Calestam, B., A Uniform Approach for Supporting Active Database Features in UML and OMT. 2001, Computer Science Department, University of Skovde: Skovde.