Framework and Architecture for the Management of Event-Condition-Action (ECA) Rule-Based Clinical Protocols

Kudakwashe Dube[1], Bing Wu[1] and Jane Grimson[2]
[1]*School of Computing, Dublin Institute of Technology, Ireland*
[2]*Department of Computer Science, Trinity College Dublin, Ireland*
E-mails: {kudakwashe.dube; bing.wu}@dit.ie, jane.grimson@cs.tcd.ie

Abstract

Computer-based support for the incorporation of clinical practice guidelines and protocol into daily practice has recently attracted a lot of research interest within the healthcare informatics area. The aim is not only to provide support for the flexible specification and execution of clinical guidelines or protocols but also the dynamic management of these guidelines or protocols. This paper presents a framework and architecture for the management of clinical protocols whose specification and execution models are based on the event-condition-action (ECA) rule paradigm.

1. Introduction

In providing computer-based support for clinical practice guidelines and protocols, the aim is not only supporting the flexible specification and execution of the clinical guidelines or protocols but also their full-scale dynamic management. A clinical *guideline* has been defined as "a set of schematic plans, at varying levels of detail, for the management of patients who have a particular clinical condition (e.g. insulin-dependent diabetes)" [1]. Clinical *protocols* are clinical guidelines, at a higher level of detail, and are usually mandatory for patients in a given clinical category [2]. In this paper the terms *clinical guideline* and *clinical protocol* are not distinguished and are used interchangeably. The ECA rule paradigm has been studied extensively in active databases. An ECA rule monitors and reacts to a situation by performing a task or an action. Situation monitoring involves detecting an event of interest and evaluating a condition associated with the event. The action is performed only if the condition holds [3]. The specification and execution of ECA rules are supported, in a limited way, in modern database systems, such as Oracle 8i, where they are commonly referred to as *triggers*. This paper presents a framework and architecture for the management of ECA rule-based clinical protocols and patient plans whose implementation is based on the ECA rule mechanism of a modern database system. Preliminary work has been presented elsewhere [4][5]. This work is part of a broad spectrum of on-going healthcare informatics research being undertaken within the MediLink Project, a national healthcare informatics research project that spans the Dublin Institute of Technology, Trinity College Dublin and several hospitals in Dublin [6]. The rest of this paper is structured as follows: Section 2 is a brief survey of related work. Section 3 presents the framework for managing ECA rule-based clinical protocols. Section 4 presents the architecture for the framework presented in Section 3. Section 5 concludes this paper.

2. Related Work

This section gives a brief survey of the computer-based support for clinical guidelines. The main focus is placed on the guideline approaches that make use of production rule formalisms. Of special interest to the authors are the approaches that make use of the ECA rule paradigm in database systems to support clinical guidelines.

Computer-Based Support for Clinical Guidelines: Several approaches have been developed for computer-based support of guideline-based care. These include:

Proceedings of the 15 th IEEE Symposium on Computer-Based Medical Systems (CBMS 2002)
1063-7125/02 $17.00 © 2002 **IEEE**

ONCOCIN [7], T-HELPER [8], Asgaard [2][9], PRO*forma* [10], the Guideline Interchange Format (GLIF) [11][12], PRESTIGE [13] and PRODIGY [14]. More recent approaches that make use of Internet technology include: the *ActiveGuidelines* model [15], which uses web-enabled connections from patient record systems to HTML-based text guidelines; and the Guideline Elements Model (GEM), which provides a generic XML-based structure for representing clinical guidelines [16][17]. These approaches use the following guideline representation formalisms and computational technologies: 1) rule-based paradigm; 2) logic-based methods; 3) network-based models; 4) workflow models; and 5) text-based formalisms [18].

Guideline Support Using ECA Rules in Database Systems: Productions rules of the form: IF *condition* DO *action,* have been used to support clinical event monitoring as well as clinical protocols [20][21]. To the best of our knowledge, besides the efforts undertaken by the authors, only two other efforts have been encountered that apply the ECA rule paradigm in supporting clinical guidelines/protocols. These efforts are: the Arden Syntax and MLMs [19]; and HyperCare [22]. These works make use of the ECA rule paradigm as defined in active database systems [3]. The Arden Syntax is a language for encoding medical knowledge bases that consists of independent modules called MLMs. MLMs are ECA rules stored as separate *text files*. Efforts have been made to build complex care plans and clinical guidelines/protocols by chaining MLMs in such a way that the action of one MLM evokes the next MLMs [25]. HyperCare is a prototype system that employs the ECA rule paradigm in the active database, Chimera, to support clinical guideline compliance in the domain of essential hypertension [22]. In HyperCare, the active capability of Chimera is used to achieve inferential capabilities of production rule expert systems.

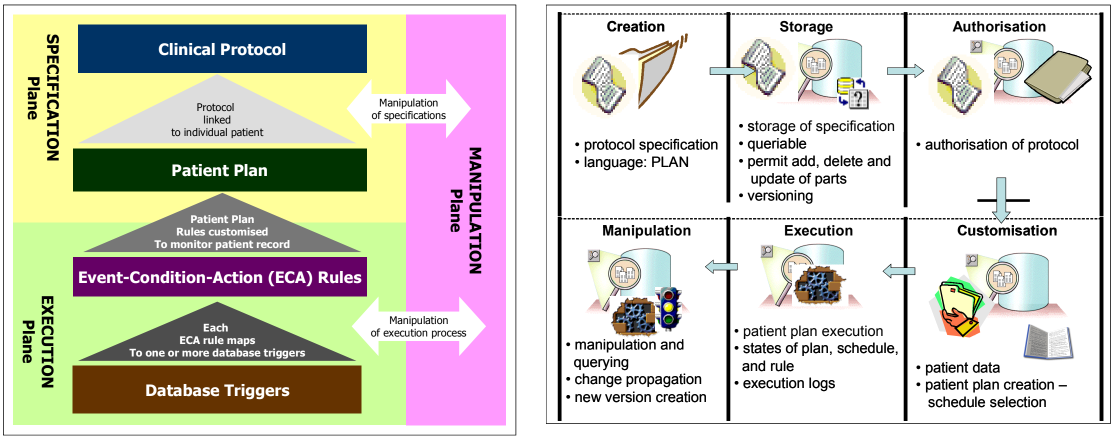
Discussion and Conclusion: One of the main reasons for the general lack of widespread use of guideline systems is the difficulty associated with integration with the electronic medical record. Integration of guideline systems with the electronic medical record allows the use of the patient's data and the presentation of guideline knowledge at the point of care while the clinician is accessing the patient's data [1]. The ECA rule paradigm has the advantage that it can be easily integrated with the electronic medical record in a database system. An important requirement is the method of representing guideline tasks using ECA rules. The Arden Syntax and HyperCare make use of the ECA rule paradigm to support clinical protocols. The former allows the generic clinical protocols to be specified and executed. Protocol specifications are stored as programming language code. There is no flexible support for the management of both specifications and their instances. HyperCare does not support the creation of generic clinical protocol specifications. Instead, the system was built for a specific clinical protocol, which it implements using ECA rules of an active database system. In the work presented in this paper, an approach that allows the management of ECA rule-based clinical protocols is taken. The approach allows generic clinical protocols to be declaratively specified, stored, executed and dynamically manipulated. Both the specification and its instances are manageable on a full-scale.

3. Framework for Managing ECA Rule-based Clinical Protocols

This section presents the framework for supporting the management of ECA rule-based clinical protocols. The management of clinical protocols and plans involves: a) the specifying of complex clinical protocols from components, b) the execution of patient care plans that are created from protocol specifications; c) the manipulation of patient care plans; d) the consideration of the state and effects of the patient care plans over time; e) the monitoring of the execution of patient care plans, and f) the issuing of queries on the static and dynamic aspects of the protocols. Support is required for: a) the specification and storage of generic protocols, b) the customization and linking of the generic protocol to the patient, thus creating a patient care plan, c) manipulation operations as well as queries against the protocol and patient care plan database, d) the ECA rule-based execution of patients care plans.

3.1. Framework for the Management of Clinical Protocols

This section presents the framework for supporting clinical protocols. Figure 1 illustrates the framework for managing ECA rule-based protocols. The framework presented in Figure 1(a), consists of the three planes: specification, execution and manipulation planes. Figure 1(b) illustrates the processes that span the three planes of Figure 1(a).


SPECIFICATION Plane
Clinical Protocol
Protocol linked to individual patient
Manipulation of specifications
Patient Plan
MANIPULATION Plane
Patient Plan Rules customised To monitor patient record
Event-Condition-Action (ECA) Rules
EXECUTION Plane
Each ECA rule maps To one or more database triggers
Manipulation of execution process
Database Triggers
Creation
• protocol specification
• language: PLAN
Storage
• storage of specification
• queriable
• permit add, delete and update of parts
• versioning
Authorisation
• authorisation of protocol
Manipulation
• manipulation and querying
• change propagation
• new version creation
Execution
• patient plan execution
• states of plan, schedule, and rule
• execution logs
Customisation
• patient data
• patient plan creation – schedule selection

(a) Information and knowledge planes for managing clinical protocols.

(b) Processes and data flow spanning the planes is diagram (a).

Figure 1. Framework for managing ECA rule-based clinical protocols

Protocol specifications are created in the *specification plane*. In the *execution plane*, the customisation of protocols produces patient care plans that suits the individual patient's condition and recent pathology. Also in the execution plane, the patient care plan is executed and execution state data is generated and made available for querying and decision-making. The protocol specifications and the running plan instances are managed in the *manipulation plane*. The interaction between the specification and the execution planes involves: 1) the *customisation* of a generic specification to suit a specific situation (patient condition); 2) the *instantiation* of a customised specification; and 3) the propagation of *dynamic changes* between the specification and the executing instance. The framework allows clinical protocols to be specified, stored, executed and manipulated both statically and dynamically. The next sections describe how this is achieved.

3.2. Specification of the Clinical Protocols

A clinical protocol specification (**Pr**) is expressed as a composition of sets of ECA rules for managing patients in a clinical problem category such as diabetes and/or its sub-types I and II. A patient plan (**Pl**) is a version of the clinical protocol that has been linked and/or customized for a particular patient. The description of the clinical protocol specification is illustrated in Figure 2. There are two basic types of rules: the *static (temporal) rule* (sr) and *dynamic rule* (dr). Static rules model actions that are to be executed either once-off or repeatedly within a period of time, as specified by t_e. Temporal rules are described as *static* because they model actions that are compulsory and whose execution time is bound on creation of the patient plan. Dynamic rules are typical ECA rules, whose execution is situation-dependent. Protocol and schedule rule sets are sets of dynamic rules whose scope is the *protocol* and the *schedule* respectively.

Entity Type	Entity	Denotation/Definition
Basic Rule Components	Event	e
	Time event	t_e
	Condition	C
	Action	A
ECA Rules	Static (Temporal) Rule	$sr = < t_e,c,a>$
	Dynamic Rule	$dr = <e,c,a>$
Rule Sets	Temporal (Static) Rule Set	$R_{sr} = U(sr_i)$
	Protocol Rule Set	$R_{dr} = U(dr_i)$
	Schedule Rule Set	$R_{sch} = U(dr_i)$
	Schedule	$s = <c, R_{sr},R_{sch}>$
	Schedule Set	$S = U(s_i)$
Protocol and Patient Plan Specifications	Patient Plan	$Pl = <X,Y>$; where $X = S - R_{sch}$ and $Y = R_{pr} \cup R_{sch}$ where $S, R_{sch}\ R_{pr}\ \varepsilon\ Pr$
	Clinical Protocol	$Pr = <S, R_{pr}, R_{st}>$
	Patient	Pt
Mappings	Patient Plan Creation	$M(Pt)\colon Pr \rightarrow Pl_{spec}$
	Patient Plan Instantiation	$I\colon Pl_{spec} \rightarrow Pl_{instance}$

Figure 2. Denotations and definitions of rules, rule set, clinical protocols and patient plan specification using the ECA rule paradigm.

A schedule is a collection of static and schedule rules that are logically related to each other and share a common goal. The schedule is part of a protocol. A schedule has some entry criteria, modeled by a condition (c), to be satisfied by the patient if the schedule is to be selected for that patient. A protocol may have more than one schedule. A protocol may also contain protocol rules. Protocol rules have no relationship with any of the protocol's schedules and models reactive behaviour for the protocol. A protocol may also contain static rules, which are not part of any schedule. A *patient plan* is created from the generic clinical protocol by the mapping M(Pt) whose effect is 1) evaluating entry criteria, 2) dropping schedules whose entry criteria do not hold for the patient and 3) customizing the rules so that they monitor the individual patient. The patient plan is instantiated through the mapping I, which generates SQL for the database triggers that implement the patient plan in the underlying database system. The database system serves the purpose of an execution engine. Currently, a declarative specification language, PLAN [4], is being used to specify such protocols.

3.3. Querying and Manipulation of the Protocol Specifications and their Instances

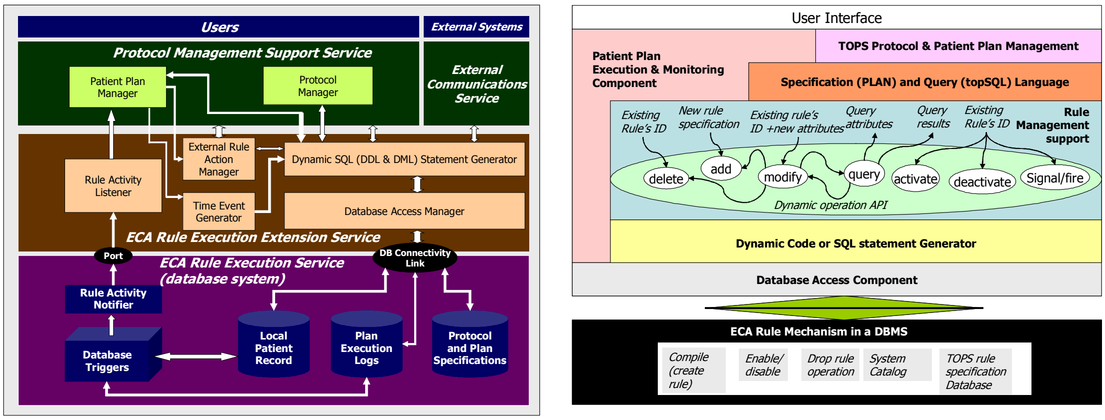
Once a protocol specification has been created, it should be stored. Once the specification is stored, it should be executable and manageable through the standard manipulation operations of addition, deletion and modification, as well as navigation and querying. The problem of version maintenance also becomes important. This section briefly discusses the querying and manipulation of the protocol and patient care plan specifications.

3.3.1. Queries: An important requirement is that the specifications, the executing instances (processes) and the effects (outputs) of the clinical protocols should be queriable. In the model presented here, the task of querying the clinical protocols is based on querying the ECA rules. Two examples of such queries are: a) Which rules refer to the (column) AGE of (table) PATIENTS in their condition? b) Which rules modify (column) DOSAGE of (table) MEDICATION in their action? This type of queries requires access to the internal structure of the rule's condition and action. The querying of rules down to component level in modern database systems is not adequately supported mainly due to the fact that rules are considered as schema objects. The use of the relational database system for storing specifications and providing the necessary extensions to the database trigger mechanism makes available the expressive power of the SQL for querying the rule-base.

3.3.2. Manipulation Operations: Another important requirement is that users should be able to manipulate (add, delete, modify, activate/deactivate, invoke, and replay execution of) the rules, rule sets and entire patient care plans at any point in time in order to support flexibility and to allow the evolution of the rule-base in the system. For this to be possible, ECA rules, which are the building blocks for creating protocol specifications, should be dynamically manageable on a full scale. Dynamic operations on schema objects, such as rules, are generally not adequately supported in modern database systems. Once again, the use of the relational database system for storing specifications and implementing extensions to the database trigger mechanism makes available the expressive power of the SQL for manipulating the rule-base.

4. Architecture for the Management of Protocols and Plans

The architecture for supporting the dynamic management of ECA rule-based protocols and patient plans is illustrated in Figure 3(a) and 3(b). The architecture allows the management of ECA rule-based clinical protocols by providing, within the framework presented in Section 3, a rule management support component that allows rule manipulation operations to be performed and queries to be issued dynamically at any time during the execution of patient plans.


Users
External Systems
Protocol Management Support Service
Patient Plan Manager
Protocol Manager
External Communications Service
External Rule Action Manager
Dynamic SQL (DDL & DML) Statement Generator
Rule Activity Listener
Time Event Generator
Database Access Manager
ECA Rule Execution Extension Service
Port
DB Connectivity Link
ECA Rule Execution Service (database system)
Rule Activity Notifier
Database Triggers
Local Patient Record
Plan Execution Logs
Protocol and Plan Specifications
User Interface
TOPS Protocol & Patient Plan Management
Patient Plan Execution & Monitoring Component
Specification (PLAN) and Query (topSQL) Language
Existing Rule's ID
New rule specification
Existing rule's ID +new attributes
Query attributes
Query results
Existing Rule's ID
Rule Management support
delete
add
modify
query
activate
deactivate
Signal/fire
Dynamic operation API
Dynamic Code or SQL statement Generator
Database Access Component
ECA Rule Mechanism in a DBMS
Compile (create rule)
Enable/ disable
Drop rule operation
System Catalog
TOPS rule specification Database

(a) Architecture for the specification, execution and manipulation of ECA rule-based clinical protocols

(b) Architectural view illustrating the support for the dynamic (on-the-fly) operations on rules.

Figure 3. Architecture for the management of ECA rule-based clinical protocols

As illustrated Figure 3(a), the architecture has four layers. The first layer consists of users and external systems. The second layer is the clinical protocol management service that allows users to specify, store, execute and manipulate clinical protocols and external systems to supply and receive information from the system. The third layer provides services that extend the ECA rule execution mechanism of the underlying database system to which the layer acts as a wrapper. The fourth layer is the ECA rule execution service.

The architecture in Figure 3 has been implemented using the Oracle 8i database system and Java. The implementation of the specification and execution planes of the clinical protocol management framework has been completed. Preliminary tests conducted ran 100 protocol instances at the same time and produced promising results.

The implementation of the manipulation and querying aspects of the clinical protocol management framework is nearing completion. A limited version of the prototype system is currently being prepared for undergoing tests using clinical protocols for blood glucose control and the diagnosis and management of micro-albuminuria in the diabetes domain at a Dublin hospital's diabetes clinic within the framework of the MediLink Project.

5. Conclusion

An important requirement in supporting clinical protocols is that these protocols must be dynamically manageable on a full-scale in order to be acceptable for routine use in daily practice. This paper has briefly described a framework and architecture for the management of ECA rule-based clinical protocols and patient plans. Supporting the management of ECA rules and their composites in a modern database system can be used as a basis for providing the flexible management of clinical protocols whose specification and execution models are based on the ECA rule paradigm. The management of the collection of ECA rules in a database system is an important and challenging requirement that can be of beneficial use in many areas within the healthcare domain as well as in other domains.

Acknowledgement: The authors would like to thank the Office of Postgraduate Studies and Research of the Dublin Institute of Technology who are sponsoring the work of K. Dube and B. Wu under its Strategic Research and Development Scheme.

References

1. Shahar Y, (2001). Automated Support to Clinical Guidelines and Care Plans: the Intention-Oriented View, Ben Gurion University, Israel, http://www.openclinical.org/resources/reference/briefingpapers/shahar.pdf, March 2002.
2. Miksch S (1999). Plan Management in the Medical Domain. AI Communications, 12(4), pp.209-235.
3. Dittrich KR, Gatziu S and Geppert A (1995). The Active Database Management System Manifesto: A Rulebase for ADBMS Features. In: Sellis T (ed): Proc 2nd Workshop on Rules in Database (RIDS), Athens, Greece, September 1995. Lecture Notes in Computer Science, Springer, 1995.
4. Wu B and Dube K (2001a). PLAN: a Framework and Specification Language with an Event-Condition-Action (ECA) Mechanism for Clinical Test Request Protocols. In Proceedings of the 34th Hawaii International Conference on System Sciences (HICSS-34): the Mini-Track in Information Technology in Healthcare, Abstracts and CD-ROM of Full Papers, IEEE Computer Society, Los Alamitos, California, p.140.
5. Wu B and Dube K (2001b). Applying Event-Condition-Action Mechanism in Healthcare: A Computerised Clinical Test-Ordering Protocol System (TOPS). Proceedings 3rd International Symposium on Cooperative Database Systems for Advanced Application (CODAS 2001), Beijing, China, April 23-24, IEEE Computer Society, Los Alamitos, California, pp.2-9.
6. MediLink (2002). The MediLink Programme, http://www.cs.tcd.ie/medilink/ ,February 2002.
7. Shortliffe, E. H., Scott, C. A., and Bischoff, M. B. (1981). ONCOCIN: An expert system for oncology protocol management. In Proc. Seventh International Joint Conference on Artificial Intelligence, pages 876--881.
8. Musen MA, Carlson RW, Fagan LM and Deresinski SC (1992). T-HELPER: Automated support for community-based clinical research. Proceedings of the 16th Annual Symposium on Computer Application in Medical Care, Washington, D.C., pp.719-723.
9. Shahar Y, Miksch S, and Johnson P (1998). The Asgaard Project: A Task-Specific Framework for the Application and Critiquing of Time-Oriented Clinical Guidelines. Artificial Intelligence in Medicine 14:29-51, 1998.
10. Fox J, Johns N and Rahmanzadeh A (1998). Disseminating medical knowledge: the PROforma Approach. Artificial Intelligence in Medicine, 14:pp.157-181.
11. Ohno-Machado L, Gennari JH, Murphy S, Jain NL, Tu SW, Oliver DE, et al. The GuideLine Interchange Format: A Model for Representing Guidelines. Journal of the American Medical Informatics Association 1998;5(4):357-372.

12. Peleg M, Boxwala AA, Omolala O, Zeng Q, Tu SW, Lucson R, Bernstam E, Ash N, Mork P, Ohno-Machado L, Shortliffe EH and Greenes RA (2000). GLIF3: The evolution of a guideline representation format. In Overhage MJ, ed, Proceedings 2000 AMIA Annual Symposium, LA, CA, Hanley and Belfus, Philadelphia.
13. C Gordon and M Veloso (1996). The PRESTIGE Project: Implementing Guidelines in Healthcare MIE 96
14. Johnson PD, Tu SW, Booth N, Sugden B and Purves IN (2000). Using scenarios in chronic disease management guidelines for primary care. In: Overhage MJ, ed, Proceedings of the of the 2000 AMIA Annual Symposium (LA, CA, USA), Hanley and Belfus, Philadelphia.
15. Tang TC and Young CY (2001). ActiveGuidelines: Integrating web-based guidelines with computer-based patient records. 2001 AMIA Annual Symposium, http://www.amia.org/pubs/symposia/D200173.PDF, March 2002.
16. Karras BT, Nath SD, and Shiffman RN 2000). A Preliminary Evaluation of Guideline Content Markup Using GEM—An XML Guideline Elements Model. In Overhage M.J.,, Ed., Proceedings of the 2000 AMIA Annual Symposium (Los Angeles, CA, 2000), 942, Hanley & Belfus, Philadelphia.
17. Shiffman RN, Karras BT, Agrawal A, Chen RC, Marenco L and Nath S (2000). GEM: A proposal for a more comprehensive guideline document model using XML. J. Am. Med. Inform. Assoc., 7:pp.488-498.
18. Tu SW and Musen MA (2001) Representation Formalisms and Computational Methods for Modeling Guideline-Based Patient Care, In: Heller B, Loffler M, Musen M and Stefanelli M (2001). Computer-Based Support for Clinical Guidelines and Protocols, Proceedings of the First European Workshop on Computer-based Support for Clinical Guidelines and Protocols, 2000, Leipzig, p.115-132, IOS Press, Amsterdam.
19. Hripscak G, Luderman P, Pryor TA, Wigertz OB and Clayton PD (1994). Rationale for the Arden Syntax. Computers and Biomedical Research, 27:pp.291-324.
20. Shortliffe, E. H., Axline, S. G., Buchanan, B. G., Merigan, T. C. and Cohen, S. N. (1973). An Artificial Intelligence Program to Advise Physicians Regarding Antimicrobial Therapy. Computers and Biomedical Research **6**: 544-560.
21. Starren, J. and Xie, G. (1994). Comparison of Three Knowledge Representation Formalisms for Encoding the NCEP Cholesterol Guidelines.Eighteenth Annual Symposium on Computer Applications in Medical Care, Washington, DC, Hanley & Belfus, Inc., 792-796.
22. Caironni PVC, Portoni L, Combi C, Pinciroli F and Ceri S (1997). HyperCare: a Prototype of an Active Database for Compliance with Essential Hypertension Therapy Guidelines. In: Proc 1997 AMIA Ann Fall Symposium, Philadelphia, PA, Hanley and Belfus, 1997:pp.288-292
23. ASTM (1992). E-1460: Standard Specification for Defining and Sharing Modular Health Knowledge Bases (Arden Syntax for Medical Logic Modules). In 1992 Annual Book of ASTM Standards, Vol.14.01, pp.539-587. American Society for Testing and Materials, Philadelphia, 1992.
24. HL7 (1999). Arden Syntax for Medical Logic Modules. Standard of the Health Level 7, 1999, http://www.cpmc.columbia.edu/arden/arden-2.1-proposed-120300.doc, March 2002.
25. Sherman EH, Hripcsak G, Starren J, Jenders RA and Clayton P (1995). Using intermediate states to improve the ability of the Arden Syntax to implement care plans and reuse knowledge. Proc Annu Symp Comput Appl Med Care 1995: pp.238-42
26. Paton NW (1999). Active Rules in Database Systems. Springer, New York.