# INTEROPERABILITY OF HETEROGENEOUS SOURCES OF PATIENT CLINICAL DATA AND MULTIPLE CLINICAL PRACTICE GUIDELINES FORMALISMS TO LEVERAGE COMPUTERIZED DECISION SUPPORT

Teză destinată obţinerii
titlului ştiinţific de doctor inginer
la
Universitatea "Politehnica" din Timişoara
în domeniul
CALCULATOARE ŞI TEHNOLOGIA INFORMAŢIEI
de către

**Ing. Valentin-Sergiu Gomoi**

| | |
|---|---|
| Conducător ştiinţific: | prof.univ.dr.ing Vasile Stoicu-Tivadar |
| Referenţi ştiinţifici: | conf.univ.dr. Seroussi Brigitte |
| | prof.univ.dr. Gheorghe-Ioan Mihalaş |
| | prof.univ.dr.ing. Horia Ciocârlie |

Ziua susţinerii tezei: 28.09.2012

România, 300159 Timişoara, Bd. Republicii 9,
tel. 0256 403823, fax. 0256 403221
e-mail: editura@edipol.upt.ro

# Acknowledgment

This thesis was possible with the partial financial support of the strategic grant POSDRU/88/1.5/S/50783 (2009) of the Ministry of Labor, Family and Social Protection, Romania, co-financed by the European Social Fund – Investing in people.

I would like to express my gratitude to my advisor, Prof. dr. eng. Vasile Stoicu-Tivadar from “Politehnica” University of Timişoara, Romania, for his uninterrupted support, valuable experience, professionalism, patience, many questions and commitment to my doctoral educational process. He had guided me through all he steps necessary in order to do the research and write the thesis.

I would like to thank Assist. prof. dr. Brigitte Seroussi from University Pierre et Marie Curie (Paris 6) of Paris for guiding me through my research stage in Paris at Tenon Hospital, for her passion for mentoring, for her review and improvement of my work, her help bringing great value to the thesis.

I want to thank to Dr. Jean-Baptiste Lamy and the staff of University of Paris 13, Paris, France, “Laboratoire d'informatique médicale et bioinformatique”, department director: Prof. dr. Alain Venot, to Dr. Harry Karadimas and Vahid Ebrahiminia MD, PhD from Henri Mondor Hospital, Biostatistics and Medical Informatics Department, Creteil, France, for their professionalism and valuable advices.

Next, my best thoughts and many thanks go to Dr. Bogdan Timar from Diabetes Clinic, Emergency County Clinical Hospital, Timișoara and to Elena Bernad MD, PhD, from the University of Medicine and Pharmacy “Victor Babeş” Timişoara, Romania, Department of Obstetrics and Gynecology, for their constant help and having the great generosity to provide medical data necessary for the validation of my work.

Great thanks to Prof. dr. Gheorghe Mihalaş, Prof. dr. eng. Diana Lungeanu from University of Medicine and Pharmacy “Victor Babeş”, Timişoara, Romania, Department of Medical Informatics, Prof. dr. eng. Lacrămioara Stoicu-Tivadar, Prof. dr. eng. Ioan Filip and S.l. dr. eng. Dorin Berian from “Politehnica” Timişoara, Romania, Department Automation and Applied Informatics, for their support, expertise and advices.

I am thanking also my colleagues from the Faculty of Automation and Computers, Vida Mihaela, Dan Dragu, Oana Lupşe and Gal Norbert for a very good collaboration and for the interesting discussions.

I would like to express my gratidude to Conf. dr. eng. Marius Bălaș and Conf. dr. eng. Valentina Emilia Bălaș from University „Aurel Vlaicu” from Arad for all the support.

Finally, I would like to thank my family and all my colleagues from the University “Politehnica” Timişoara, Romania.Teza de doctorat a fost elaborată pe parcursul activităţii mele în cadrul Departamentului de Automatică al Universităţii „Politehnica” din Timişoara.

Timişoara, 09.2012 Valentin-Sergiu Gomoi

Gomoi, Valentin-Sergiu

**Interoperability of heterogeneous sources of patient clinical data and multiple clinical practice guidelines formalisms to leverage computerized decision support**

Teze de doctorat ale UPT, Seria 14, Nr. 9, Editura Politehnica, 2012, 127 pagini, 47 figuri, 8 tabele.

ISSN:2069-8216
ISSN:2069-8216
ISBN:978-606-554-541-0

Cuvinte cheie: clinical decision support systems, virtual medical record, clinical practice guideline,HL7 standards, Arden Syntax, Clinical Document Architecture*.*

Rezumat,
The main subject of the thesis is the improvement of clinical decision support systems (*CDSS*) by developing patient data and medical rules acquisition methods. The ultimate goal is to increase the level of interoperability of clinical decision support systems.

Main research objectives:
- Developing of clinical decision support (*CDS*) architecture that allows the integration of multiple data sources types based on *HL7* standards.
- Developing and describing methods for the generation of new medical rules with the help of the *data mining* and their translation in *Arden Syntax* formalism in order to increase the medical rules base and CDSS interoperability.
- Increasing *CDSS* interoperability by acquisition of medical rules in a format accepted as standard from different nonstandard formalisms.
- Validation of the proposed solutions.

# TABLE OF CONTENTS

# LIST OF ACRONYMS

| | |
|---|---|
| arff | Atribute Relation File Format |
| ATC | Anatomical Therapeutic Chemical |
| BMI | Body Mass Index |
| CCD | Continuity of Care Document |
| CCR | Continuity of Care Record |
| CDS | Clinical Decision Support |
| CDSS | Clinical Decision Support System |
| CPG | Clinical Practice Guidelines |
| DAM | Domain Analysis Model |
| DeGeL | Digital Electronic Guideline Library |
| EHR | Electronic Health Record |
| GLIF | GuideLine Interchange Format |
| HIPAA | Health Insurance Portability and Accountability Act |
| HL7 | Health Level 7 |
| HL7 CDA | Health Level 7 Clinical Document Architecture |
| INN | International Nonproprietary Name |
| ISO | International Standards Organization |
| LA | Logic of Argument |
| LOC | Logic of Obligation and Time |
| MLM | Medical Logic Module |
| OCL | Object Constraint Language |
| RIM | Reference Information Model |
| SOAP | Simple Object Access Protocol |
| UML | Unified Modeling Language |
| UMLS | Unified Medical Language System |
| vMR | virtual Medical Record |
| XML | eXtensible Markup Language |

# LIST OF FIGURES

# LIST OF TABLES

# 1. INTRODUCTION

Studies have shown that the use of Clinical Practice Guidelines (*CPGs*) result in the improvement of patient quality of care by reducing practice variability or care costs [1]. *CPGs* are documents with recommendations and instructions to assist medical professionals and patients in decision making, based on results of scientific research followed by discussion and expression of expert-opinions, to make effective and efficient medical practice [2]. A clinical protocol can be defined as a local adaptation of one or more clinical guidelines; it provides for instance the information on duration, dosages, procedures which are often not specified in *CPGs* [3].

*CPGs* and medical protocols are initially developed as narrative documents. But, they proved to be unable to change physician behaviors when disseminated in textual format. However, when implemented in care points as computer-based clinical decision support systems (*CDSS*s), they may increase physician decision compliance with recommendations [4, 5]. One definition for Clinical Decision Support is: "Clinical Decision Support systems link health observations with health knowledge to influence health choices by clinicians for improved health care" [6]. Studies regarding statistical data in the establishment of these support systems in healthcare are presented in [7]. One of the statistics shows that the use of *CDSS* increased physicians' performances in 64% of the studies and regarding the patient outcomes, 13% of studies established a benefit. In [8], it is stated that in over 90% randomized controlled trials, clinical practice improved, based on the use of *CDSS*.

Beside the already known advantages of narrative protocols and guidelines, other advantages can be added when implementing medical decision support systems: faster implementation of new medical knowledge, integration of many local databases, decreasing in costs, better protocol visualization in a graphical manner, avoiding reading vast amounts of data regarding each step of a narrative medical protocol and so on [2]. Another definition of a *CDSS* has been proposed by Hunt [9]: "A *CDSS* is defined as any software in which characteristics of individual patients are matched to a computerized knowledge base for the purpose of generating". Thus, to develop a *CDSS* we need coded patient data, a computerized structured knowledge base modeling the domain expertise necessary to solve diagnostic or therapeutic patient-specific problems, and software solutions to match knowledge to data and generate patient-specific recommendations.

The work carried during my PhD research covered three dimensions of *CDS* (Clinical Decision Support): retrieve patient data, elaborate a knowledge base, and reasoning based on patient data and medical knowledge, to generate best recommendations.

Concerning patient data retrieval, I have worked on existing standards such as *HL7 CDA* (Health Level 7 *Clinical Document Architecture*) [10] and *HL7 vMR* [11] which have been proposed to enable the communication between different medical units systems. In order to reduce the gap between medical knowledge representations (as medical rules in: *ASTI*, *Asbru, PROforma, SAGE, Gaston* etc. [1, 12, 13, 14, 15]) and medical data sources (e.g., patient data) an architecture which allows the use of multiple *HL7 CDA*s as patient data source might improve the adaptability of *CDS* to local context.

Also other solution might be the use of the *HL7 vMR* standard in conjunction with *HL7 CDA* documents in order to make the *CDSS*s more adaptable by extending the medical data source used for the inference.

Multiple solutions for clinical decision support systems (*CDSS*) have been developed in the last years in order to improve healthcare act. Several of these solutions are: *Asbru, PROforma, Degel, ASTI, Gaston, Egadss*, [1, 16, 17, 18, 19] etc.

Concerning the patient data retrieval, a major problem of the existing approaches is the difficulty of integrating local data bases [20]. One *CDS* solution which is developed in a hospital would be very difficult to be deployed in another medical unit. To solve this problem, standards as *HL7* have to be integrated in the new solutions that will be developed. Approaches as *Gello* or *Egadss* already implement several related standards [21, 22].

Although there are solutions which offer a degree of interoperability (based on standards) with other medical informatics systems, there are major problems for the implementation of this solution in local medical units. For example, *Egadss* is one of the solutions which offer a high degree of interoperability compared to other solutions (like: *ASTI*, GASTON, *PROforma*, *Asbru*). *Egadss* uses two *HL7* standards: *Arden Syntax* [23] for the representation of medical rules and *HL7 CDA* [10] for the representation of patient data. The implementation of these standards is not enough; next paragraphs will present the issues which are still to be resolved.

*Egadss* has as patient data entry only one *HL7 CDA* document, coming from a unique patient data source. In this manner it can be used only for a local data repository. This can lead to an incomplete image of the patient status. In order to solve this problem it is necessary to implement the integration of information from multiple data sources (laboratory, other medical units, radiography, and so on) using *HL7 CDA* or even other standards (*HL7 vMR*).

*CPGs* have been developed to provide recommendations based on current evidence for best practices in the management of a given disease. Since *CPGs* are elaborated from the results of scientific publications; they only give solutions for patient profiles that correspond to patient eligibility criteria of clinical trials. Strength of guidelines is evaluated by the quality of the research studies they come from. The grade of evidence of a guideline has the highest value (A) when it comes from a randomized clinical trial. On the other hand, when no research studies are available, therapeutic propositions can be provided by a consensus of experts. However, not all patient profiles are covered by *CPGs* [24]. In the special case of the management of hypertension, the coverage of patient profiles as described by Seroussi et al [25] was 8% with evidence-based guidelines, and 47% with expert consensus-based advices. As a consequence, *CPGs* provide no solution for about 45% of patient profiles. For these patients, one can complete the knowledge base with expert-based propositions or use knowledge discovery from databases to learn how to manage the profiles which lack *CPGs* support.

Concerning the gap in the *CPGs* which lead to the impossibility to offer solutions in some situations, we have worked on the development of a module which allows discovery of medical knowledge by using *data mining* tools. This module regards the mapping of medical rules obtained using *data mining* rules to *Arden Syntax* formalism. The implementation of *Arden Syntax* allows the use of the new discovered knowledge in multiple medical units (which implement *Arden Syntax* standard). The use of *data mining* and *Arden Syntax* might increase:

- the acceptance of *CDSS* by offering solutions which were not covered by the evidence-based guidelines and expert consensus-based advices;
- the adaptability of the new medical rules by the implementation of *HL7* standards.

The use of multiple formalisms for the representation of medical knowledge results in difficulties obtaining *CDSS* interoperability. *Asbru, ASTI* and *Gaston* are solution which use their own model for the representation of medical knowledge, which makes almost impossible the use, in different medical units, of knowledge represented with the help of a specific formalism (in one particular medical unit). The use of a formalism that is accepted as a standard (*Arden Syntax*) for the representation of medical rules may allow the use of the same medical knowledge in different medical units. One solution to enable the transmission of medical knowledge between the different solutions is the mapping of the knowledge contained by these solutions to *Arden Syntax*.

Concerning the translation in *Arden Syntax* of medical knowledge represented with other formalisms, I have analyzed the expressiveness of *Arden Syntax* compared with object oriented formalism (*ASTI*) for the representation of *CPGs*. I also proposed mapping between *ASTI* and *Arden Syntax*. Since the medical rules represented in *ASTI* need specific patient treatment information, another concern of this research was the way in which the needed patient data can be extracted from the *vMR* data model (which is the proposed standard for the representation of patient medical information for *CDSS*) in order to make the reasoning.

In Fig. 1.1, the main components of a system which generate medical recommendations and advices are depicted. My doctoral studies depicted the three modules at the bottom of the Fig. 1.1: the patient data retrieving and the medical knowledge gathering and representation.

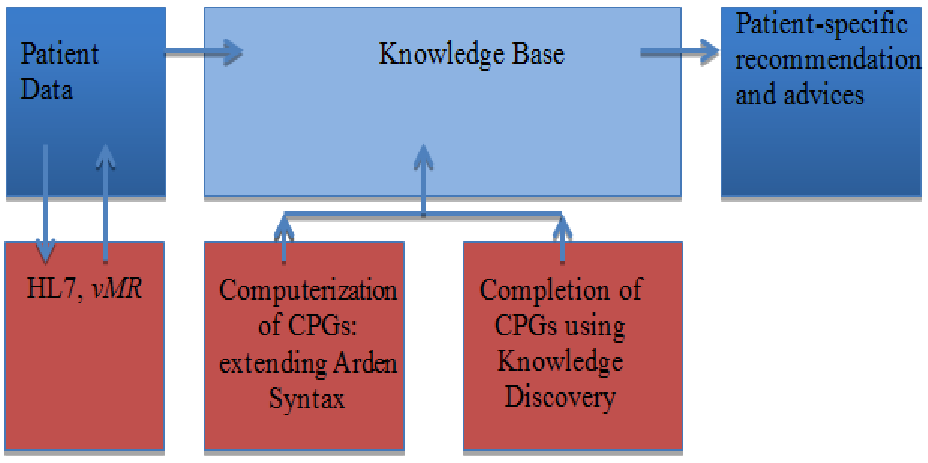

Patient Data
Knowledge Base
Patient-specific recommendation and advices
HL7, vMR
Computerization of CPGs: extending Arden Syntax
Completion of CPGs using Knowledge Discovery

Fig. 1.1. Clinical patient data, formalisms for representing knowledge *CPGs* and discovery of knowledge, components for the generation of patient specific recommendations and advices

## 1.1. Challenges regarding patient data retrieving

Medical recommendations (generated based on inference) become more precise as the data about a patient is more complex [26]. In order to have as many as possible data about a patient, multiple data sources need to be integrated in clinical decision support systems. The integration of many local databases represents an advantage for clinical practice, by gathering more relevant clinical information. Putting together information from different sources has the inconvenient that data is stored in many formats. A solution created and implemented in a certain medical unit is very difficult to be deployed in other units.

### 1.1.1. Research direction - patient data retrieving based on standards

The implementation of standards regarding the representation of patient data and the representation of medical rules can be a valid solution for the deployment of *CDSS* in different medical units.

Also, the *HL7 Group* proposed several standards for the representation of patient data [27, 28, 29]:

- *HL7 CDA* (Health Level 7 Clinical Document Architecture)
- *HL7 CCD* (Health Level 7 Continuity of Care Document)
- *HL7 CCR (*Health Level 7 Continuity of Care Record*)*
- *HL7vMR* (Health Level 7 *virtual Medical Record*)

Several standards where proposed for the representation of medical rules, by *HL7* group:

- Arden Syntax
- Gello

In order to increase the interoperability of the current approaches regarding the generation of medical recommendation, I developed a solution which implements several of the above mentioned standards, which will be further presented in the thesis:

- HL7 CDA (Health Level 7 Clinical Document Architecture)
- HL7vMR (Health Level 7 virtual Medical Record)
- Arden Syntax

## 1.2. Problems related to the computer implementation of *CPGs* and knowledge discovery

There is no unique formalism for the representation of medical knowledge. The development of institution specific formalisms resulted in the impossibility of medical knowledge exchange between different medical units. Impossibility to exchange medical knowledge may lead to the duplication of the same medical knowledge in different formats, resulting in:

- more specialists needed for the translation of *CPGs* in computer format;

- longer time needed;
- increase of costs.

Beside the problems related to the interoperability, another issue of computer implementation of *CPGs* is the lack of recommendations for all the possible clinical situations. One solution for this problem is the development of *knowledge discovery* modules.

### 1.2.1. Research directions: Knowledge discovery, translation of medical rules obtained with data mining to *Arden Syntax*

Adding new knowledge in medicine can improve the medical act. The use of *data mining* in collaboration with clinical decision support systems can bring new knowledge in *CDS*. *Data mining* technology is used to generate medical rules based on the local practice, and to add new knowledge to *CDSS*s. In order to integrate the *data mining* rules with different *CDSS*s, I developed a tool which translates these rules to *Arden Syntax* formalism. Medical rules obtained with *data mining* are not evidence-based, the results obtained by using this tool are patient specific advices.

By translating of *data mining* rules into *Arden Syntax*, the new rules can be also exported in other medical units which implement *Arden Syntax* standard.

## 1.3. Challenges regarding the representation of *CPGs*

Regarding the knowledge representation of the *CPGs*, numerous solutions using almost the same number of unique formalisms, are developed and implemented.

In order to model *CPGs*, two main techniques emerged [30]:

- Rule-based technology: Arden Syntax;
- Task Network: Asbru, PROforma, EON, GLIF.

The *Arden Syntax* is a clinical guideline formalism accepted as an official standard by *HL7*; it is a textual language, intuitive and without room for ambiguity. It is freely available, a mature and actively maintained open standard [21].

### 1.3.1. *Task Network* technology

The use of *Task Network* technology allows a more suitable representation of *CPGs*. In order to represent the *CPGs* there are used sets of tasks. These tasks differ from one approach to the other and they can be: preferences, intentions, conditions, effects, decisions, actions, branch, synchronization, etc. (e.g. use of tasks Fig. 1.2). These are usually referred as ontology of tasks. This technique can describe relations between the components of the plan and usually also provide tools for visual representation of plans and organization of tasks within the plan. The Task Network has responded to a number of requirements [30, 31]:

how to interpret situations and events;
the best choice of clinical action;
a way of reflecting the clinical tasks and constraints.

Several of the most known approaches which use *task network technology* are:

Asbru
PROforma
GLIF

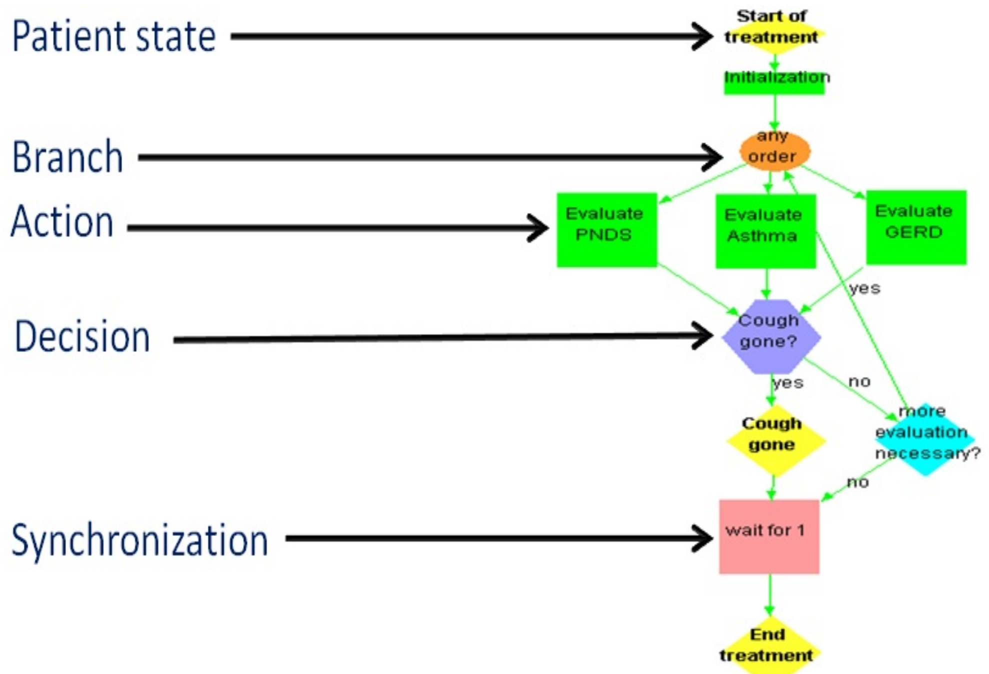


Fig. 1.2. Representation of medical guidelines with *GLIF* Model, adapted after [32]

These approaches are not as widely used as *Arden Syntax.* The use of such formalisms is not suited for an increased interoperability between different clinical decision support systems.

### 1.3.2. Research directions: using *Arden Syntax* as standard for the representation of medical knowledge.

The interoperability of the *CDSS*s can be improved by using *CPGs* representation formalism which is accepted as a standard. As presented in 1.3.1, *Arden Syntax* has been proposed as standard formalism by *HL7* group, in order to allow the exchange of medical knowledge between different institutions. The use of *Arden Syntax* in a *CDSS* increases usability of such a system by allowing the access to a more vast medical knowledge source.

The translation of existing medical rules, represented in different formalisms, to *Arden Syntax* formalism can result in standardized representation of medical knowledge, which can be interchanged between different medical units.

Analyzing the translation of medical rules represented in a different medical formalism to *Arden Syntax* may be considered the first step for the acquisition of medical rules from other sources than narrative guidelines. This analysis focuses on the expressiveness of *Arden Syntax* and the way in which more sophisticated concepts (as decision, branch, treatment, actions) from other formalisms (as *Asbru*, *PROforma* or *GLIF*) can be represented in *Arden Syntax*.

I analyzed the representation of *ASTI* concepts in *Arden Syntax* and I propose several mappings between the two concepts. This mapping is realized in order to allow the acquisition (in a format accepted as standard) of medical knowledge directed from a "formalism" instead of using the narrative guideline as source of medical rules, this way the number of medical specialists in the process of knowledge formalization is reduced.

Concerning the representation of different medical rules from different formalisms, attention should be paid to the needed patient data. Each approach needs specific patient data. I try to identify how this information can be represented in the *vMR* model.

## 1.4. Objectives

In order to respond to the mentioned challenges, the next objectives are proposed:

A. Analysis of the current state of the clinical decisions support systems development and the identification of new research directions
B. Developing a *CDS* architecture that allows the integration of multiple data sources types based on *HL7* standards
C. Developing and describing methods for the integration of *Topic Maps* technology and the *vMR* standard, in *CDSS*s.
D. Developing and describing methods for the generation of new medical rules with the help of the *data mining* and their translation in *Arden Syntax* formalism in order to increase the medical rules base.
E. Increasing *CDS* interoperability by acquisition of medical rules in a format accepted as standard from different nonstandard formalisms.
F. Evaluation of the proposed solutions quality.

## 1.5. Thesis structure

In order to achieve the previous objectives the next Chapters were defined:

***Chapter 1*** – In this part are introduced several of the most actual challenges in the development of Clinical Decision Support systems. The medical context of healthcare quality is presented regarding the *CDS*. Different problems are identified concerning three areas of *CDS*:

- retrieving patient specific data;
- knowledge representation;
- knowledge discovery.

Several solutions are introduced in order to solve the above mentioned problems, having as result the improvement of clinical decision support systems.

***Chapter 2*** – An analysis regarding different clinical decision support systems is presented. This analysis concerns about: the formalism used for the representation of medical rules, acquisition of patient data, and visualization of the medical recommendation. Differences between these *CDSS*s are highlighted. The history of different approaches in the domain is presented, as well. The visualization part of the clinical recommendations (at the creation time an execution) is emphasized.

***Chapter 3*** – Contains the presentation of a *CDS* architecture that is developed in order to allow the access of the *CDS* to more complex medical data and medical knowledge. For the development of this architecture several standards were considered: *vMR HL7*, *HL7 CDA*, *Arden Syntax*. The architecture has as main components:

- Data manger – used for connecting all other parts;
- an inference engine;
- different tools for collecting patient specific data;
- an interface for the representation of medical recommendation and patient data.

***Chapter 4*** – Presents a solution for the situations when computerized *CPGs* cannot offer a medical recommendation. This solution is based on the use of *data mining* technology for extracting medical rules from data. These medical rules can be used to offer medical advices. In order to extend the use of these new rules a convertor is developed.

This convertor translates the medical rules obtained with *data mining* techniques in *Arden Syntax* format, which is recognized as a standard formalism. This system was tested for the data concerning newborns from the *Obstetrics and Gynecology department of Bega clinic*, *Timișoara,* for advices regarding the *Apgar* score.

***Chapter 5*** – Contains an analysis regarding the representation of medical rules from *ASTI* formalism in *Arden Syntax* and retrieving of the needed patient data, from the *vMR*, in order to make the inference. Based on this analysis several solutions are presented for the representation of *ASTI* medical rules in *Arden Syntax*. In this way instead of implementing the medical rules in *Arden Syntax* from *CPGs*, they can be translated directly from *ASTI* formalism reducing the number of personnel needed.

In order to make the inference based on the *ASTI* rules the *vMR* data model (patient data representation standard for *CDS*) was considered as the patient data source. The analysis revealed the actual *vMR* data model do not allow the representation of all patient data needed for the inference. This is why I suggested classes that should extend the *vMR* data model.

***Chapter 6*** – Summarizes discussions and conclusions on the research, and emphasizes the contributions and future development suggestions that the author has made to improve the clinical decision support systems.

# 2. STATE OF THE ART

Implementation of evidence based recommendations in different institutions has as results [33]:

- Improving quality of care,
- Reducing variation in medical practice,
- Dissemination of scientific results,
- Protects practitioners in terms of malpractice,
- Structuring medical information,
- Consensus among physicians,
- Elimination of uncertainties,
- Increased confidence of the medical personnel in the medical act undertaken,
- Integration of services and new procedures,

Furthermore the implementation of medical protocols and guidelines by using computers, promises to improve acceptability and applicability in daily clinical practice, the physician being able to monitor medical activities. Another advantage of using computerized medical protocols is the integration with the local context, so, by using the local databases the decision support system can generate context specific recommendations (e.g. based on: the number of nurses, existing medical devices, existing products in stock, and so on). The use of clinical decision support systems can lead to faster implementation of new medical practices, standardization of practices and / or an intuitive graphical interface to track the steps of a protocol in a much easier way (opposite to narrative protocols) [1, 34]. Computer implementation of guidelines and protocols brings also other advantages (to increase the compliance of medical staff with *CDS* [5, 35, 36]) such as:

- possibility to generate alarms (e.g. due to sudden changes in the data collected from the patient);

possibility to choose from several treatments,

- possibility to insert references and links for documentation, after each recommendation.

This Chapter has the role to highlight the necessity of clinical decision support systems, the different challenges as well as to review solution in this domain. A short presentation of the several solutions in the domain of clinical decision support is made. Various comparisons are made in order to see which of these solutions is the most suited for the development of clinical decision support with a high degree of interoperability. In order to enhance the interoperability, there are also analyzed, the standards for the representation of patient data.

## 2.1 Standards for the representation of patient clinical data

In this, the *HL7* standards for the representation of patient clinical data and the representation of medical rules are presented.

### 2.1.1. *HL7*

*HL7* is an open international standard for communication which allows independent development of medical information systems and automatic communication between these systems. The purpose of the standard is to allow the exchange, management and integration of data that support clinical patient care, management, delivery and evaluation of health services.

Benefits of using *HL7* are: allowing information to be exchanged between different software applications, reduced paperwork, costs reduction and providing flexibility. It can be implemented using a variety of software technologies in order to suit the requirements of the developer [37, 38].

### 2.1.2. *HL7 vMR*

Without a common model representation of information, the effort required for mapping information between different institutions becomes almost useless. Moreover, different models of semantic information may be incompatible and unable to be mapped between different systems.

As stated in [8, 39, 40], important causes for limitation of implementing *CDS* in medical units are: the use of different models, the lack of a standard representation of clinical information and terminologies associated that are used in medical institutions. To meet these needs, the Working Group *HL7 CDS vMR* initiated the *vMR* project, which had as objective to support the development of scalable and interoperable *CDS*, by establishing a "standard model to represent clinical information inputs and outputs that can be transmitted between systems *CDS* and other medical systems." [41].

This model contains 131 (Table 2.1) medical data elements. The *vMR* data model appeared as a necessity for the interoperability between different *CDS* and data sources. This data model allows the representation of a large range of information concerning the patient. In the development of the *vMR* model (the patient data and the requirements to be integrated) 22 institutions from 4 different countries have been involved (representing 20 *CDSS*s). *vMR* data model is in process of becoming a standard for the representation of medical knowledge used in different *CDSS*s in order to solve the problem of interoperability of *CDSS*s [41].

| Data Element | Examples | % Systems Using DE* |
|---|---|---|
| Demographic Data Elements | | |
| Patient Gender | Male | 95% |
| Patient Race(s) | Black | 55% |
| Patient Birth Date | February 19, 1975 | 75% |
| Patient Age | 45 years | 75% |
| Patient Age Group | 19+ years | 55% |
| Postal Address(es) | 100 Main St., Cary, NC | 40% |
| P*RIM*ary Care Provider | Dr. Jenkins, Clinic X | 55% |
| Moved out of Area | True | 11% |
| Encounter Data Elements | | |
| Location Type Code(s) | SMD code for ICU | 65% |
| Encounter Location | Health System A Clinic X | 75% |
| Provider Type Code(s) | *HIPAA* nephrology code | 50% |
| Encounter Status | Completed, Missed | 60% |
| Date/Time Interval | 3/1/08 to 3/2/08 | 65% |
| Encounter Identifier | Encounter ID ABCDEF | 45% |
| Encounter Note(s) | Discharge summary | 55% |
| Procedure Data Elements | | |
| Procedure Code | SMD code for biopsy | 85% |
| Procedure Site Code | SMD code for right breast | 40% |
| Procedure Modifier Code | CPT laterality modifier | 45% |
| Date/Time Interval | 3/5/08 3:15 – 7:40 pm | 70% |
| Procedure Status | Active, Canceled | 32% |
| Procedure Identifier | EHR Entry # 1234567 | 35% |
| Associated Enc. ID | Encounter ID ABCDEF | 25% |

| Data Element | Examples | % Systems Using DE* |
|---|---|---|
| Adverse Reaction Observation Data Elements | | |
| Causative Agent Type | Medication, Food | 70% |
| Causative Agent Code | SMD code for lisinopril | 65% |
| Agent Class(es) | ACE inhibitor | 47% |
| Reaction Code | SMD code for weal | 55% |
| Reaction Severity | SMD code for severe | 65% |
| Reaction Date/Time | Early 1980s | 50% |
| Reaction Status | Active, Inactive | 32% |
| Laboratory Result Observation Data Elements | | |
| Test Type | Chemistry, Pathology | 75% |
| Test Status | Ordered, Completed | 75% |
| Test Code | LNC code for HgbA1c | 90% |
| Specimen Location Code | SMD code for lungs | 65% |
| Specimen Type Code | SMD code for sputum | 65% |
| Collection Date/Time | 3/15/08 3:15pm | 85% |
| Value | 135 mg/dL | 80% |
| Normal Range | 50 mg/dL – 150 mg/dL | 70% |
| Interpretation | Panic High | 95% |
| Nested Observations | Hgb & HCT in CBC | 55% |
| Observer Identifier | Laboratory XYZ | 26% |
| Observation Note | Report contents | 35% |
| Note Type | Pap test report | 21% |
| Physical Finding Observation Data Elements | | |
| Finding Type | Vital sign, radiology | 75% |
| Finding Code | SMD code for SBP | 75% |

| Data Element | Examples | % Systems Using DE* |
|---|---|---|
| Procedure Note | Report contents | 35% |
| Procedure Note Type | Colonoscopy report | 26% |
| Data Elements Common to All Types of "Observations" Below | | |
| Observation Identifier | EHR Entry # 1234567 | Ave. 35% |
| Observation Date/Time | 3/5/08 3:15 pm | Ave. 58% |
| Observer Type | MD, RN, Patient | Ave. 22% |
| Associated Enc. ID | Encounter ID ABCDEF | Ave. 33% |
| Problem Observation Data Elements | | |
| Observation Type | Problem List Observ. | 65% |
| Problem Code | *ICD9* code for diabetes | 95% |
| Problem Class(es) | Cardiovascular disease | 26% |
| Problem Modifier | Negative/does not have | 32% |
| Problem Status | Active, Resolved | 65% |
| Status Time Interval | 1995 to present | 55% |
| Observation Method | Histological confirmation | 5% |
| Medication Observation Data Elements | | |
| Observation Type | Prescription, Usage | 70% |
| Medication Code | SMD code for lisinopril | 100% |
| Medication Class(es) | ACE inhibitor | 67% |
| Medication Dose | 30 mg, 2 puffs | 70% |
| Medication Route | PO, IV, IM | 70% |
| Medication Rate | BID prn, 12mg/hr, qam | 75% |
| Coverage Time Interval | 11/1/07 to 3/1/08 | 60% |
| Refill Information | On 2nd of 6 total refills | 50% |
| Medication Status | Active, Inactive | 63% |
| Patient Position Code | SMD standing code | 40% |
| Finding Location Code | SMD code for lungs | 40% |
| Value | 125 mm Hg | 76% |
| Normal Range | 90 – 140 mm Hg | 50% |
| Interpretation | Normal, High | 55% |
| Nested Observations | SBP & DBP within BP | 50% |
| Finding Status | Active, completed | 16% |
| Finding Note | Report contents | 29% |
| Note Type | Cardiac exam note | 11% |
| Goal Observation Data Elements | | |
| Goal Focus Code | SMD code for SBP | 40% |
| Value | 135 mg/dL | 40% |
| Other Observation Data Elements | | |
| Observation Type | Social History, Survey | 70% |
| Obs. Focus Code | LNC for survey inst. | 65% |
| Value | 5 packs/day, true | 73% |
| Interpretation | Normal, abnormal | 45% |
| Nested Observations | Items in survey | 45% |
| Patient Affiliation Data Elements | | |
| Affiliated Entity Type | Insurer, Care Provider | 45% |
| Entity Identifier | Medicaid, Clinic X | 40% |
| Obs. Date/Time | 3/15/08 | 35% |
| Affiliation Status | Active, Inactive | 35% |
| Status Time Interval | 3/15/08 – 3/15/09 | 30% |
| *CDS* Context Data Elements | | |

| Data Element | Examples | % Systems Using DE* |
|---|---|---|
| Medical Equipment Observation Data Elements | | |
| Observation Type | Prescription, Usage | 20% |
| Equipment Code | SMD code for wheelchair | 25% |
| Family History Observation Data Elements | | |
| Relationship to Patient | SMD code for aunt | 45% |
| Relative Demographics | 57 year old female | 30% |
| Relative Age of Death | N/A, 85 years | 30% |
| Relative Problem(s) | Problem info as above | 35% |
| Relative's EHR data | Mother's EHR record | 5% |
| *CDSS* User Type | Physician, Patient | 42% |
| User Preferred Language | English, Spanish | 26% |
| Info Recipient Type | Physician, Patient | 37% |
| Info Recipient Language | English, Spanish | 47% |
| Task Context | Order entry, lab review | 35% |
| Data Elements for All Orderable Items (e.g., Meds, Labs) | | |
| Orderable Item Status | Ordered, Completed | 37% |
| *CDS* Resource Data Elements | | |
| Concept Taxonomy | *ICD9* codes for COPD | 58% |

Table 2.1. *vMR* data elements [41]

In Fig. 2.1 a UML (Unified Modeling Language) representation of the *vMR* is depicted. Here we can see that each patient has clinical statements associated.

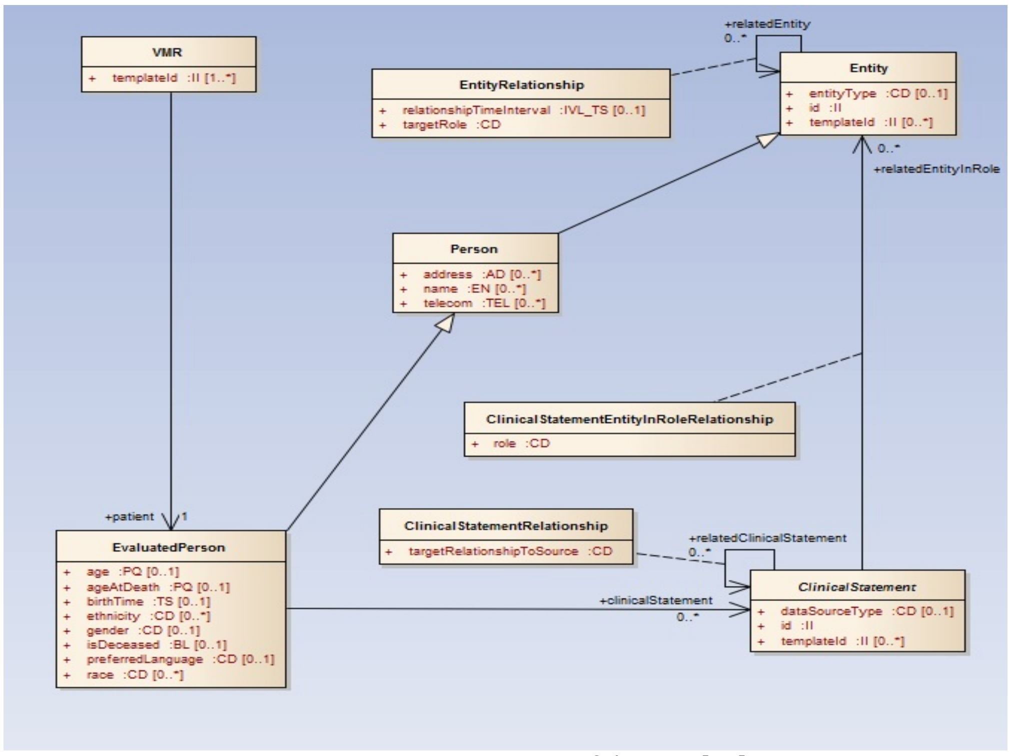


Fig. 2.1. UML representation of the *vMR* [28]

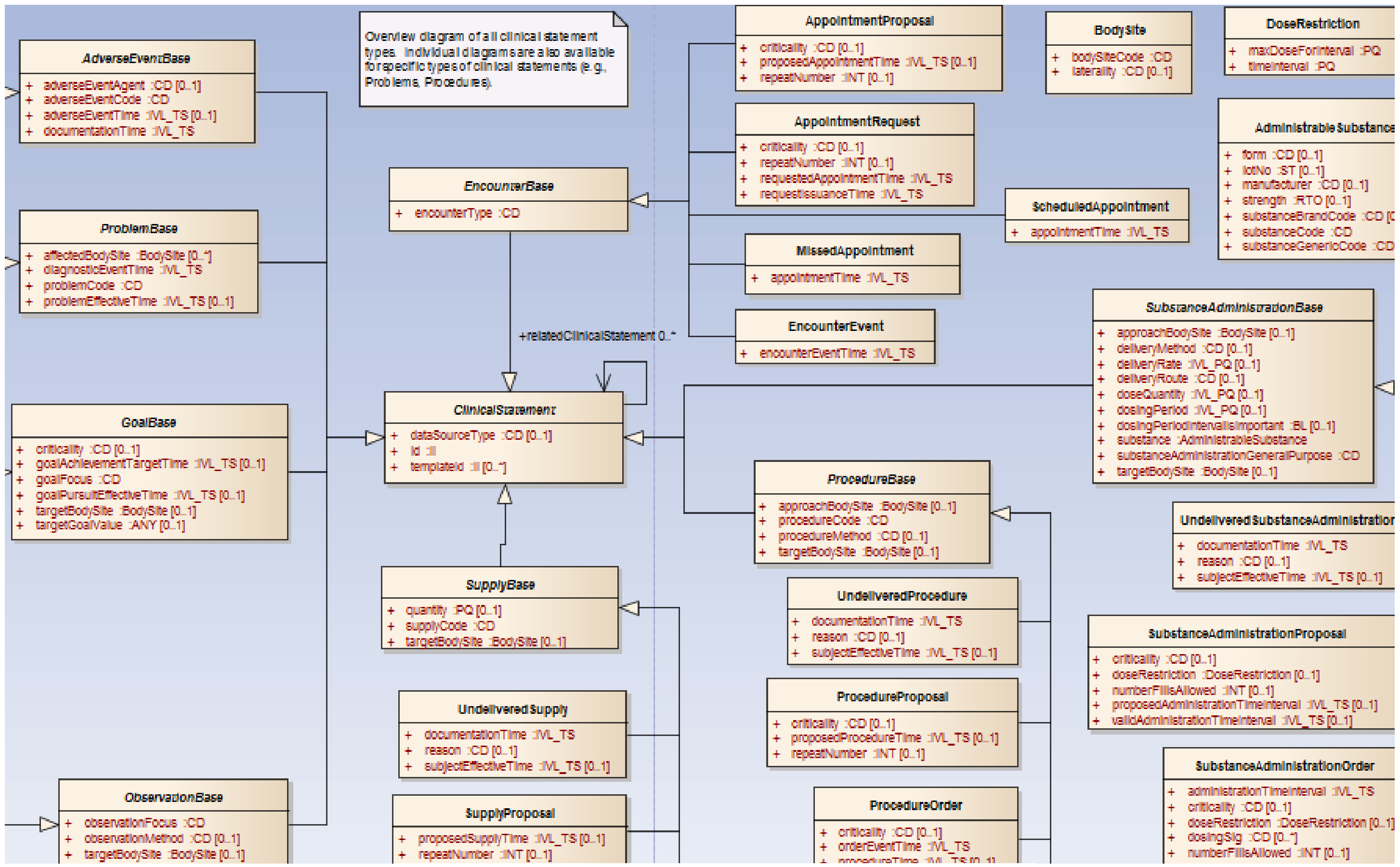

Overview diagram of all clinical statement types. Individual diagrams are also available for specific types of clinical statements (e.g., Problems, Procedures).
AdverseEventBase
+ adverseEventAgent :CD [0..1]
+ adverseEventCode :CD
+ adverseEventTime :IVL_TS [0..1]
+ documentationTime :IVL_TS
ProblemBase
+ affectedBodySite :BodySite [0..*]
+ diagnosticEventTime :IVL_TS
+ problemCode :CD
+ problemEffectiveTime :IVL_TS [0..1]
GoalBase
+ criticality :CD [0..1]
+ goalAchievementTargetTime :IVL_TS [0..1]
+ goalFocus :CD
+ goalPursuitEffectiveTime :IVL_TS [0..1]
+ targetBodySite :BodySite [0..1]
+ targetGoalValue :ANY [0..1]
ObservationBase
+ observationFocus :CD
+ observationMethod :CD [0..1]
+ targetBodySite :BodySite [0..1]
EncounterBase
+ encounterType :CD
+relatedClinicalStatement 0..*
ClinicalStatement
+ dataSourceType :CD [0..1]
+ id :II
+ templateId :II [0..*]
SupplyBase
+ quantity :PQ [0..1]
+ supplyCode :CD
+ targetBodySite :BodySite [0..1]
UndeliveredSupply
+ documentationTime :IVL_TS
+ reason :CD [0..1]
+ subjectEffectiveTime :IVL_TS [0..1]
SupplyProposal
+ proposedSupplyTime :IVL_TS [0..1]
+ repeatNumber :INT [0..1]
AppointmentProposal
+ criticality :CD [0..1]
+ proposedAppointmentTime :IVL_TS [0..1]
+ repeatNumber :INT [0..1]
AppointmentRequest
+ criticality :CD [0..1]
+ repeatNumber :INT [0..1]
+ requestedAppointmentTime :IVL_TS
+ requestIssuanceTime :IVL_TS
ScheduledAppointment
+ appointmentTime :IVL_TS
MissedAppointment
+ appointmentTime :IVL_TS
EncounterEvent
+ encounterEventTime :IVL_TS
BodySite
+ bodySiteCode :CD
+ laterality :CD [0..1]
DoseRestriction
+ maxDoseForInterval :PQ
+ timeInterval :PQ
AdministrableSubstance
+ form :CD [0..1]
+ lotNo :ST [0..1]
+ manufacturer :CD [0..1]
+ strength :RTO [0..1]
+ substanceBrandCode :CD [0
+ substanceCode :CD
+ substanceGenericCode :CD
SubstanceAdministrationBase
+ approachBodySite :BodySite [0..1]
+ deliveryMethod :CD [0..1]
+ deliveryRate :IVL_PQ [0..1]
+ deliveryRoute :CD [0..1]
+ doseQuantity :IVL_PQ [0..1]
+ dosingPeriod :IVL_PQ [0..1]
+ dosingPeriodIntervalIsImportant :BL [0..1]
+ substance :AdministrableSubstance
+ substanceAdministrationGeneralPurpose :CD
+ targetBodySite :BodySite [0..1]
ProcedureBase
+ approachBodySite :BodySite [0..1]
+ procedureCode :CD
+ procedureMethod :CD [0..1]
+ targetBodySite :BodySite [0..1]
UndeliveredProcedure
+ documentationTime :IVL_TS
+ reason :CD [0..1]
+ subjectEffectiveTime :IVL_TS [0..1]
ProcedureProposal
+ criticality :CD [0..1]
+ proposedProcedureTime :IVL_TS [0..1]
+ repeatNumber :INT [0..1]
ProcedureOrder
+ criticality :CD [0..1]
+ orderEventTime :IVL_TS
UndeliveredSubstanceAdministratio
+ documentationTime :IVL_TS
+ reason :CD [0..1]
+ subjectEffectiveTime :IVL_TS [0..1]
SubstanceAdministrationProposal
+ criticality :CD [0..1]
+ doseRestriction :DoseRestriction [0..1]
+ numberFillsAllowed :INT [0..1]
+ proposedAdministrationTimeInterval :IVL_TS [0..1]
+ validAdministrationTimeInterval :IVL_TS [0..1]
SubstanceAdministrationOrder
+ administrationTimeInterval :IVL_TS
+ criticality :CD [0..1]
+ doseRestriction :DoseRestriction [0..1]
+ dosingSig :CD [0..*]
+ numberFillsAllowed :INT [0..1]

Fig. 2.2. UML representation of Clinical Statement [28]

Clinical statement (Fig. 2.2) can be one of the following types:

- goal –aim, or clinical end to be achieved;
- problem – the difficulties and clinical conditions that need to be treated;
- adverse events – event that have bad consequences regarding the patient;
- observation – used for notifying different facts;
- encounters for an evaluated person;
- procedures – representing the tasks requested to be done to accomplish a clinical aim;
- substance administration;
- supplies – represent needed equipment and other clinical material for a certain patient (example: syringe) [28, 42].

This model is not mandatory to be used as the unique data source for *CDS* but it can be used for the interoperability of the existing system. Other representations (e.g., *HL CDA*) of medical data can be used in order to make the *CDS* more adaptable to a local context. The *vMR* will be represented by using *Topincs* (a *Topic Maps* open-source software) [43]. The access to medical data will be made (through the *Data Manager*) with different services already existing in this software and new ones which are to be developed.

### 2.1.3. *HL7 CDA (Clinical Document Architecture)*

*Clinical Document Architecture* (*CDA*) can be defined as a complete information object that can exist outside of a messaging context and/or be used in an *HL7* message [10].

*HL7 CDA* is a standard that allows specifying the structure and semantics of clinical documents in order to exchange data. The *HL7 CDA* documents can include text, pictures, sounds, and other media. Data could be any of the following data: summaries, referrals, laboratory results, medical history, physical examination, diagnostic report, prescription, or general health report [10].

The *CDA* is an *XML* document that consists of a header and a body:

- Header includes: patient information, author, creation date, document type, provider,
- Body includes: admission details, diagnosis, patient details, and medications.

An example of an instance for *CDA* document is presented below:

```
<ClinicalDocument XMLns="urn:HL7-org:v3"
XMLns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <id root="111" extension="01" assigningAuthorityName="UPT"></id>
  <code code="SYSTEm" codeSystem="222"></code>
  <effectiveTime value="20110103"/> <!--yyyymmdd -->
  <author>
   .........................
  </author>
  <custodian>
    <assignedCustodian>
```

```
        <representedCustodianOrganization>
            <id/>
            <name>Dr. Ionescu</name>
        </representedCustodianOrganization>
    </assignedCustodian>
  </custodian>
      <!--Demographics-->
  <recordTarget>
     <patient>
         <id/>
         <patientPatient>
         <administrativeGenderCode code="F"
codeSystem="2.16.840.1.113883.5.1"></administrativeGenderCode>
         <birthTime value="19701109"/>
         <maritalStatusCode code="S"
codeSystem="2.16.840.1.113883.5.2"></maritalStatusCode>
         </patientPatient>
     </patient>
   </recordTarget>
  <component>
      <structuredBody>
          <!-- Active Problems section-->
          <activeProblemsComponent>
          <section>
              <code code="11450-4" codeSystem="2.16.840.1.113883.6.1"></code>
              <title>Active Problems</title>
              <entry>
                   <observation classCode="OBS" moodCode="EVN">
                      <code code="250.x" codeSystem="2.16.840.1.113883.6.2"
codeSystemName="ICD-9-CM" displayName="Diabet"/>
                  </observation>
             </entry>
          </section>
          </activeProblemsComponent>
```

In Fig. 2.3 the abstraction layers for message transmission are presented (after [44]).

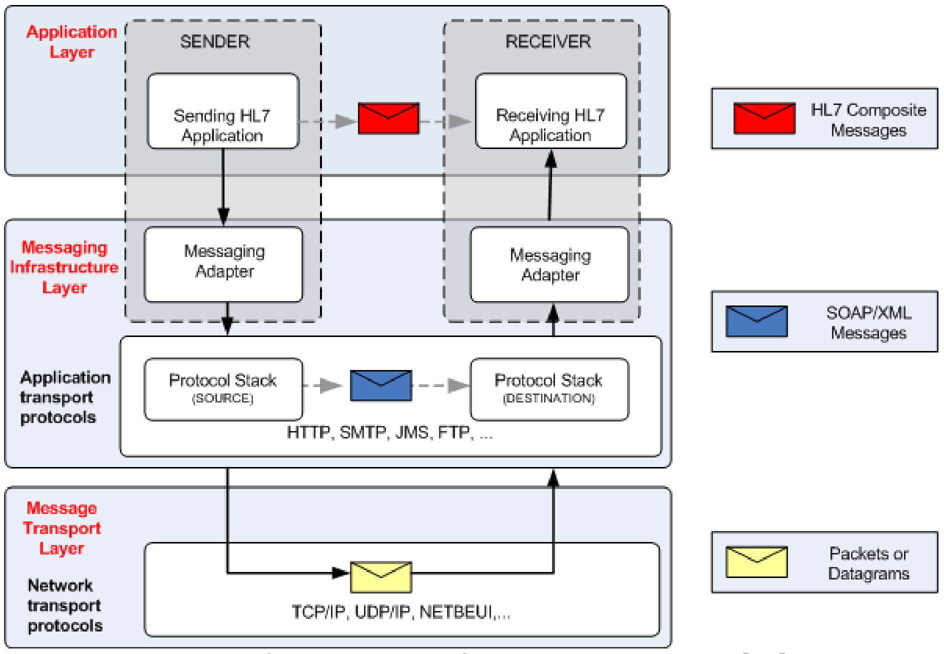

Application Layer
SENDER
RECEIVER
Sending HL7 Application
Receiving HL7 Application
Messaging Infrastructure Layer
Messaging Adapter
Messaging Adapter
Application transport protocols
Protocol Stack (SOURCE)
Protocol Stack (DESTINATION)
HTTP, SMTP, JMS, FTP, ...
Message Transport Layer
Network transport protocols
TCP/IP, UDP/IP, NETBEUI,...
HL7 Composite Messages
SOAP/XML Messages
Packets or Datagrams

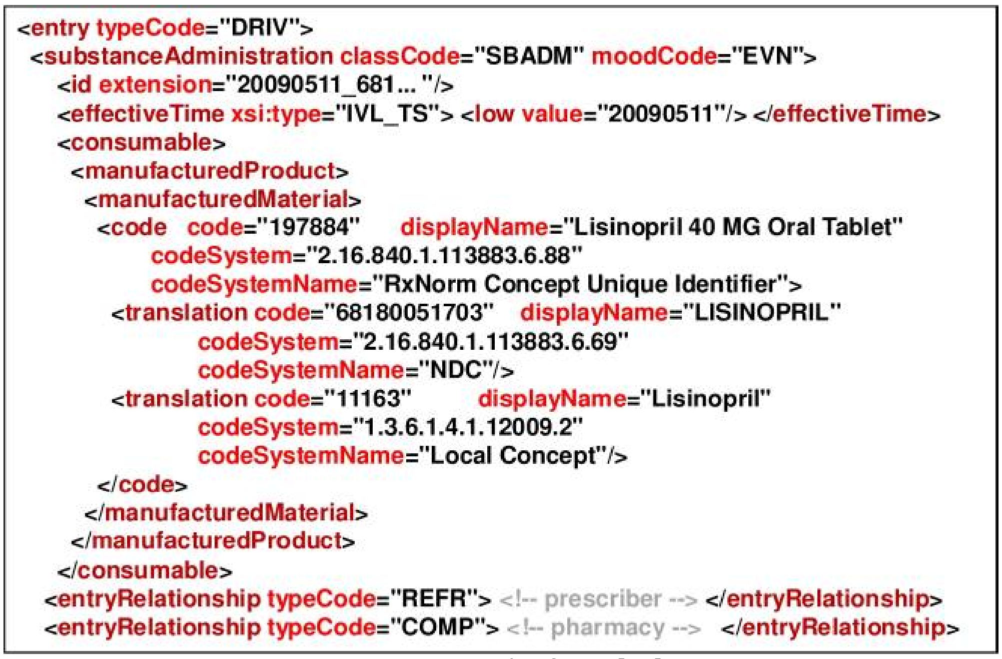
Fig. 2.3. Abstraction Layers for message transmission [44]

*SOAP* messages are used in order to encapsulate the *HL7 CDA XML* and helps the data transfer.

### 2.1.4. *CCD (Continuity of Care Document)*

*Continuity of Care Document* (Continuity healthcare document) is a communication standard used to represent information about the patient and to allow this information to be shared among providers and within personal health records. It summarizes the most common information needed about current and past state of health, in a form that can be passed to other applications [45].

*CCD* templates include: header, purpose, problems, procedures, family history, social history, payers, advance directives, alerts, medications, immunizations, medical equipment, vital signs, functional states, results, encounters and plan of care [45]. In Fig. 2.4 is presented an example of *CCD* format:



```
<entry typeCode="DRIV">
 <substanceAdministration classCode="SBADM" moodCode="EVN">
   <id extension="20090511_681... "/>
   <effectiveTime xsi:type="IVL_TS"> <low value="20090511"/> </effectiveTime>
   <consumable>
    <manufacturedProduct>
     <manufacturedMaterial>
      <code   code="197884"     displayName="Lisinopril 40 MG Oral Tablet"
          codeSystem="2.16.840.1.113883.6.88"
          codeSystemName="RxNorm Concept Unique Identifier">
       <translation code="68180051703"    displayName="LISINOPRIL"
              codeSystem="2.16.840.1.113883.6.69"
              codeSystemName="NDC"/>
       <translation code="11163"        displayName="Lisinopril"
              codeSystem="1.3.6.1.4.1.12009.2"
              codeSystemName="Local Concept"/>
      </code>
     </manufacturedMaterial>
    </manufacturedProduct>
   </consumable>
  <entryRelationship typeCode="REFR"> <!-- prescriber --> </entryRelationship>
  <entryRelationship typeCode="COMP"> <!-- pharmacy -->  </entryRelationship>
```

Fig. 2.4. Example of *CCD* [29]

The use of *CCD* brings a number of advantages:

- shared syntax;
- easy representation of data;
- shared model - providing extensibility;
- better monitoring;

## 2.2. Standards for the representation of medical rules

Beside the standards for the representation of patient data, a major role in the interoperability of clinical decision support systems is played by the standards used for the representation of medical rules.

The *HL7* group defined two standards for the representation of medical rules: *Arden Syntax* and *Gello*

### 2.2.1. *Arden Syntax*

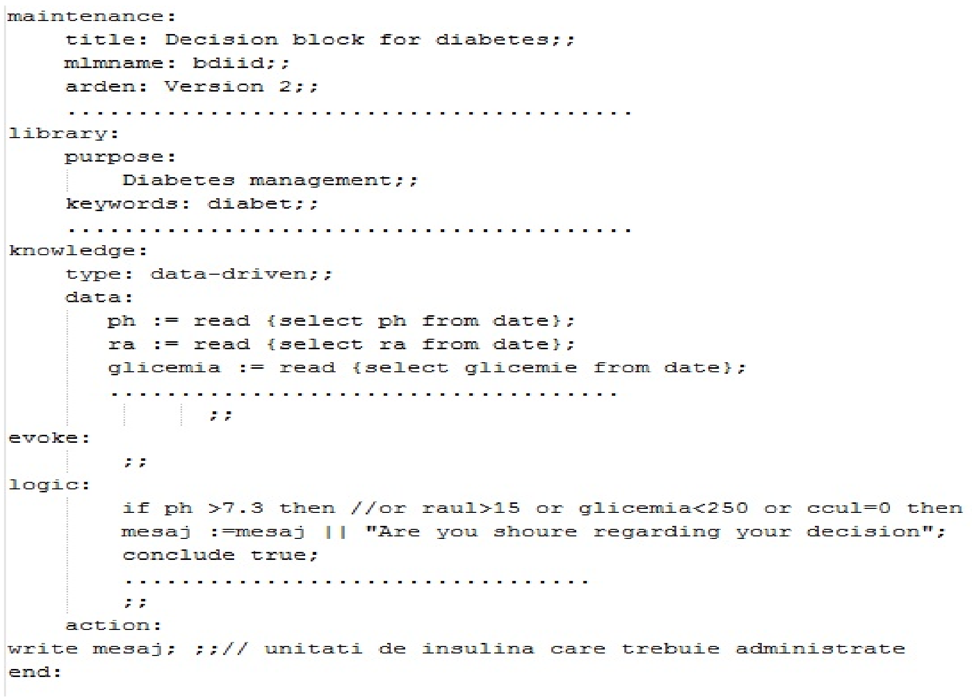
In *Arden Syntax*, medical rules are represented in *medical logical module (MLM)* [46]. Each *MLM* (Fig. 2.5) is composed of four categories: *Maintenance*, *Library, Knowledge* and *Resources* (which is optional) [27].*MLM* contain information about: the organization that created it, the date, links to other information, medical rules for taking a single medical decision [23, 47].

In *Maintenance* are presented information like: the title of the *MLM*, *MLM* name, *Arden Syntax* version, institution, author, date, etc.

In *Library* are presented: the purpose, explanation, keywords, etc.

*Knowledge* category is the most important part of a *MLM* and contains: medical rules, medical data to be extracted for inference, action to be take based on the rules. All this information is represented in 7 slots [27]:

*type* – mandatory – specifies other slots that are contained in the knowledge category;
*data* – mandatory – specifies the needed variables which are used in the *MLM* and the place from where this variables should be extracted (from the local databases);
*evoke* – mandatory – specifies the conditions (e.g., occurrence of an event) for which the *MLM* is triggered;
*logic* – mandatory – it contains the actual medical rules. Based on these rules and the data obtained in the data slot, it concludes true or false (to execute or not an action);
*action* – mandatory – it is executed when the logic slot concludes true. Actions can be: display a message, evoke other *MLMs*, etc.;
*priority* – not mandatory – specifies the priority for a *MLM* (when multiple *MLMs* satisfy the evoke criteria in the same time) to be executed and is a number from 1 (low priority) to 99 (high priority);
*urgency* – not mandatory – specifies the importance of the action of the *MLM* and is a number from 1 (low priority) to 99 (high priority)[27].


```
maintenance:
    title: Decision block for diabetes;;
    mlmname: bdiid;;
    arden: Version 2;;
    .......................................
library:
    purpose:
        Diabetes management;;
    keywords: diabet;;
    .......................................
knowledge:
    type: data-driven;;
    data:
       ph := read {select ph from date};
       ra := read {select ra from date};
       glicemia := read {select glicemie from date};
       ...................................
              ;;
evoke:
        ;;
logic:
        if ph >7.3 then //or raul>15 or glicemia<250 or ccul=0 then
        mesaj :=mesaj || "Are you shoure regarding your decision";
        conclude true;
        ................................
        ;;
    action:
write mesaj; ;;// unitati de insulina care trebuie administrate
end:
```
Fig. 2.5. *MLM* elements (example)

The *Arden Syntax* formalism is the most widely used formalism for the representation of medical rules. Only in the last five years, several of solutions which implement this formalism were developed, such as: *Egadss* [16], *Arden2ByteCode* [48], Medexter [49].

### 2.2.2. *Gello*

*Gello* is a standard for the development of clinical decision support systems. It was developed under the coordination of *HL7 Clinical Decision Support Technical Committee*. *Gello* is an object-oriented language built on existing standards. *Gello* is based on *Object Constraint Language (OCL)* developed by *Object Management Group*. In order to offer an adequate manipulation of clinical data only *OCL* relevant components were selected and integrated into *Gello* [50, 51].

*Gello* language can be used for [52]:

- creation of expression to extract and manipulate data from medical databases;
- implementation of decision criteria by building expressions to reason about data;
- creating expressions, formulas for other applications.

*Gello* was designed based on an execution model of medical guidelines proposed by *HL7 Clinical Decision Support Technical Committee*. The model consists of a series of steps: actions, decisions, status of a patient, branches and synchronization (based on *GLIF*). *Gello* provides a standard interface for medical data systems and other data or knowledge sources. The object-oriented approach provides flexibility and extensibility that is necessary to implement a wide range of applications [52].

The use of *Gello* [51, 52]:

- provides support for implementing a platform independent object oriented data model compatible with version 3 of *HL7 RIM*, and eliminates the need of methods designed to extract information;
- simplifies the creation and updating of clinical data objects and their evaluation;
- facilitates the exchange of documents (containing rules) for clinical decision support systems.

## 2.3. Existing solutions in the domain of *CDS*

Domain analysis reveals a variety of approaches related to generation of medical recommendations based on clinical guidelines and protocols. Several of the most known are: *Asbru, PROforma, EON, TSNet, Gello, Arden, Protege, Degel, ASTI, GASTON, SEBASTIAN, Arden Syntax* [1, 22, 53, 54, 55, 56, 57, 58, 59, 60, 61]. Each of these is described in terms of certain mathematical theories and methods of implementation and visualization of medical protocols and guidelines [62].

Based on these methods and mathematical theories, in order to implement, create and execute the medical guidelines and protocols on computers, each approach assumes the existence of two main modules, which allow:

- formalization of medical knowledge (“authoring engine”) by defining the steps of protocol (or defining medical rules based on clinical guidelines) and the implementation of steps in a machine language by health professionals and IT specialists;

- medical rules execution ("execution engine"), using also patient data processing, to generate medical recommendations based on medical rules described by different formalisms [63, 64, 65, 66];

Another important part of *CDS* is the visualization of medical protocols ("visualization interface"), allowing the graphical visualization of the steps and protocols and the related patient specific data, equipment, etc. [30]. This part is described in section 2.4.

In the next subsections, several solutions are presented. These solutions are object oriented and they present different components for the representation of the different steps of medical guidelines or protocols.

### *A. Asbru*

*Asbru* is a time-oriented, intention-based, skeletal plan-specification representation language that is used in the *Asgaard Project* to represent clinical guidelines and protocols in *XML*. It expresses clinical protocols as skeletal plans that can be instantiated for every patient. Initially designed for the set of plan-management tasks, it enables the designer to represent both the prescribed actions of a skeletal plan and the knowledge roles required by the various problem-solving methods for performing the intertwined supporting subtasks. *Asbru* distinguishes between a declarative data abstraction part and a procedural hierarchy of plans. The data abstraction specification consists of a set of parameter definitions, while each of the plans in the plan hierarchy consists of a name, a set of arguments, a time annotation, preferences, intentions, conditions, effects, and plan body (which is mandatory) [67, 68].

Preferences constrain the applicability of a plan to achieve a certain goal. Examples of preferences are select-method, a matching heuristic to determine the applicability of the entire plan, resources, a specification of forbidden or obligatory resources (e.g., a patient has a high blood pressure but also he has diabetes, therefore a drug to reduce blood pressure which also increases blood sugar is prohibited), and the applied strategy.

Intentions are used to model the purpose of the plan, which is independent of the plan body. As an example, intentions are useful in the selection of the most appropriate plan by defining which patient state(s) must hold during or after a plan's execution or which actions should take place. Intentions are modeled as temporal patterns [1].

To change the state of a plan, temporal patterns named *Conditions* are used. In *Asbru*, plans are in a certain state: activated, suspended, aborted and completed. So, they are defined as numbers of condition categories: "filter-preconditions", "setup-preconditions" "suspend-conditions", "abort-conditions" and "completed conditions" [69].

To select the most appropriate plan can be used effects by describing the expected behavior of the plan's execution. A treatment plan which might increase the blood-pressure level, is not the most adequate for a certain patient.

The plan body is a set of subplans or actions (which are plans that do not contain any subplans anymore) that have to be performed whenever the plan is considered appropriate (based on the plan's preconditions, intentions or effects). The order in which the subplans are executed is determined by the Plan's type and subtype attributes. In *Asbru*, the following three types of synchronizing subplans:

sequentially, parallel, anyorder, are defined. Plans can be suspended, aborted or completed (based on the plan's conditions) [1, 68].

There are various software solutions used for implementing this approach:

- authoring tools: DELT/A (Vienna), URUZ (Israel);
- visualization tools: AsbuView, CareVis (Vienna);
- guideline execution tools: *Asbru* Interpreter (Vienna), SPOCK (Israel) [56].

***B.PROforma***

*PROforma* was developed at Cancer Research UK for the general purpose of building decision support and intelligent agents. The technology includes the *PROforma* language, a formal specification language (the term used in software engineering), a knowledge representation language (as understood in artificial intelligence) and a set of Prolog and Java tools for building applications in the language. *PROforma* is essentially a first-order logic formalism extended to support decision making and plan execution, but it also incorporates a number of well-known features of non-classical logics (e.g. modal logic, temporal logic) and two novel logics (LA, logic of argument and LOT, logic of obligation and time) to support decision making and action control [57].

The approach includes a suite of authoring and execution software that incorporates also verification tools.

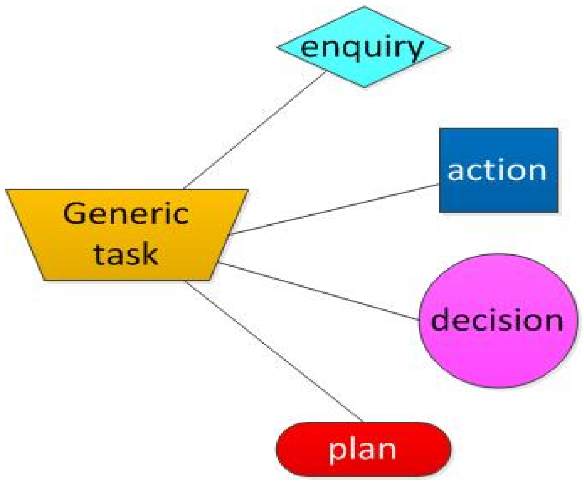


Fig. 2.6. *PROforma* ontology

Clinical guidelines and protocols are defined by using a set of tasks that can be composed into networks representing plans or procedures carried out over time (Fig. 2.6) [57]. These tasks can be:

- Plans: are a set of tasks which are carried out to achieve a clinical goal. They can include an indefinite number of tasks (any type) and other sub-plans ordered to reflect the constraints (e.g., logical, temporal).

- Decisions: are tasks which involve choices of some kind, such as a choice of investigation, diagnosis or treatment. The *PROforma* specification of a decision task defines the decision options, relevant information and a set of argument rules which determine which of the options should be chosen according to current data values [70].
- Actions: are clinical procedures that can be enacted by external agents (e.g., administration of a drug).
- Enquiries: are used to obtain various data from users or various databases. It has specification that includes the description of the information required and a method to obtain it.

### *C.GLIF*

The *GuideLine Interchange Format (GLIF*) was developed to model guidelines in terms of a flowchart that consists of structured scheduling steps, representing clinical actions and decisions. In order for the guidelines to be readable by humans, interpretable by computers and adaptable by different (local) institutions, *GLIF* defines the guideline at three levels of abstraction: conceptual, computable, and implementable level [1, 71].

#### *The conceptual level*

The highest level is the conceptual one, where guidelines are represented as flowcharts, which can be viewed by humans (e.g., guideline authors) but are not interpretable by decision support systems. At this level, details such as the contents of patient data elements, clinical actions and guideline flow are not formally specified [69].

#### *The Computable Level*

If at conceptual level the guidelines were visualized as flowcharts, at computable level the guidelines are specified using *GLIF* TNM. The new model is object-oriented. In this model the guideline tasks are described by classes. Guidelines are modeled as instances of these classes. Each class contains attributes as: administrative, describing capabilities. The guideline steps are also represented by classes. In *GLIF* there are five classes defined to represent the steps: Decision, Patient state, Branch, Synchronization and Action steps [32]. These are described as:

- Decision step directs flow from one guideline step to various alternatives. Decision steps can be classified as Case step and Choice step. A Case step is a Decision step where a number of criteria of the case condition are evaluated, and based on the outcome; the guideline flow is directed to the next step. Choice step represent the situations where guidelines present options and leaves the choice to him/her.

- A Patient state step is used as a label to describe the current patient state in a certain moment of the guideline execution. Each Patient state step contains attributes that describe the state of the patient.

- Branch steps model a set of concurrent steps and tasks with arbitrary execution order by directing flow to multiple parallel guideline steps. A Branch step has multiple subsequent steps while other types of steps have only one.

- Synchronization step is the place where the multiple guideline steps, that follow a Branch step, converge. When a synchronization step is executed, its continuation criterion is evaluated. If the continuation criterion is not satisfied, the synchronization step will continue to wait until the completion of other steps eventually leads to the fulfillment of the continuation criterion [32].

- Action step represents actions that should be performed. There are defined three types of actions: medically oriented actions (e.g., what type of drug is needed), data actions (to retrieve specific patient data from a local repository), subguideline actions.

*The Implementable Level*

Finally, at the implementable level, guidelines can be custom-tailored to particular institutional information systems. At this stage, institution-specific procedures and mappings (which are usually non-sharable) are specified, along with the formalization from the above-mentioned medical knowledge layer. Both the medical knowledge and the implementable layer are still under development [69]. *GLIF* defines the patient data and medical terminology in accordance with standard such as *HL7*'s Reference Information Model (*RIM*) or the Unified Medical Language System (UMLS).

Authoring and evaluation tools were created using Protégé knowledge environment. The tool runs by executing Protégé and loading the *GLIF* ontology [54].

The above presented solutions (*GLIF, Asbru, Gello, PROforma, Arden Syntax*) are several of the most known and appreciated in the domain of clinical decision support systems [7].

### 2.3.1. Evolution and history of existing approaches

Beside the presented solutions, other multiple approaches to generate medical recommendation have been developed in the last years. Some of them evolved during time, some just proved to be not complex enough to be further used. In Fig. 2.7 is presented a short history regarding the evolution of different approaches. This Fig. is adapted after [1], some new important approaches and tools (which emerged after 2008) being added:

- *ASTI* [72]
- *ASTI*2, *ASTI*3 – which were developed at the same time [58, 73]
- *Egadss* – which uses *Arden Syntax* as formalism
- *Arden2ByteCode* – which uses *Arden Syntax* as formalism [48]

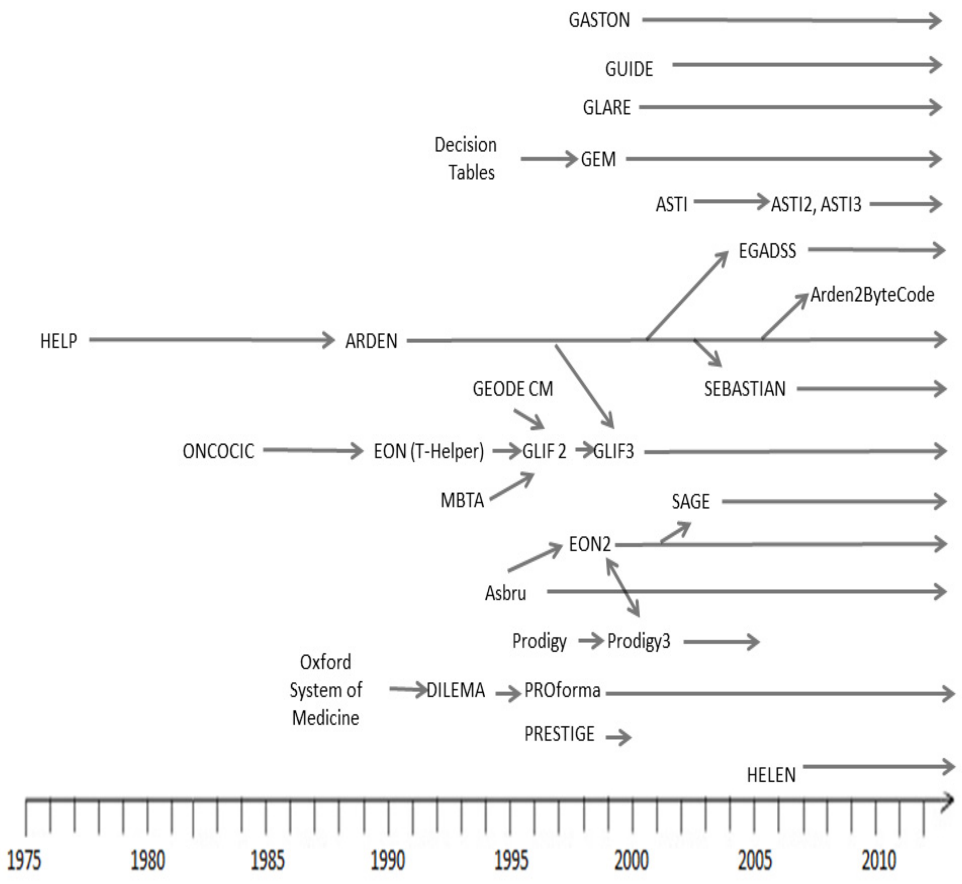


Fig. 2.7. History of the evolution of different approaches in computer-based guidelines and protocols (adapted after [1])

### 2.3.2. Comparing different solutions

Usually, these solutions have been developed to respond at particular needs. There is no solution which implemented in a local context is able to answer to all needs. The biggest problem concerning this is the lack of implementing standards for *CDS* and for the local medical information systems.

In order to address this issue, I tried to define a set of features which should be present in clinical decision support systems in order to be suited for as many situations as possible. Not all of them are needed for a singular situation. These features are:

- use of standards for the patient data inputs – important because it can allow the system to communicate with multiple medical information systems which implement standards for the representation of patient data;

- use of standards for the output of the system – it is important because it allows the system to interact with other *CDSS*s and medical information systems which implement the same standards;
- use of standards for medical rules representation –important because it allows the interchange of medical knowledge and avoid the duplication of the same medical knowledge in different formats;
- Use of *Task Network* – this technology is more suited for the representation of complex guidelines [1];
- Use of rule *Based Technology* – is easy understood by physicians;
- Knowledge discovery – important in the situations in which the evidence based rules do not cover all the clinical situations and the implementation of knowledge discovery such as *data mining* should lead to the discovery of new medical rules.

In Table 2.2 are illustrated multiple *CDS* solutions and which of the above features are supported by these solutions. The approaches to be evaluated where selected based on:

- the impact they had in the literature
- the documentation available regarding the different aspects which were evaluated.

Based on the Table 2.2 three important conclusions can be drawn:

- The presented solutions do not implement *knowledge discovery*.
- Few solutions use standards for the patient data inputs and outputs; this is one of the most important causes of for low interoperability of clinical decision support systems.
- A small number of solution use standards for medical rules representation; this is an important causes of low interoperability of clinical decision support systems, as well.

| | *Egadss* (*Arden Syntax*) | *Arden2ByteCode* (*Arden Syntax*) | *Asbru* | *ASTI 3* | *Gello* | Gaston |
|---|---|---|---|---|---|---|
| Standards for the patient data inputs | YES (*HL7 CDA*) | NO | NO * | NO* | YES | NO* |
| Standards for the output of the system | YES (*HL7 CDA*) | NO | NO* | NO* | YES | NO* |
| Task Network | NO | NO | YES** | NO | YES*** | YES*** |
| Rule-Based technology | YES (*Arden Syntax*) | YES (*Arden Syntax*) | NO | YES | NO | NO |
| Standards for medical rules representation | YES (*Arden Syntax*) | YES (*Arden Syntax*) | NO | NO | YES | NO |
| Knowledge discovery | NO | NO | NO | NO | NO | NO |

Table 2.2. Features implemented by different clinical decision support approaches

* - developed for unique medical information systems

** - based on an ontology of tasks

*** - allows the definition of tasks

## 2.4 *CDS* user interfaces

The purpose of this part is to present different methods and software to enable visualization of medical guidelines and protocols at various levels: design, testing and implementation. Various methods and software are used for many years in the medical systems. Unlike these applications used for the visualization of X-ray, 3D scanning and tomography, the tools for visualization of patient data, treatments, laboratory result or computerized medical guidelines has not been improved as much lately[74].

Regarding the visualization of medical guidelines, one can notice two main directions: visualization during the computer formalization of guidelines and the visualization of medical recommendations and medical data during execution. The first direction refers to the process of creating computer-based guidelines, where the focus is on presenting the various components of the plan (structure guidelines) to medical experts. The second direction refers to the representation of clinical guidelines, during execution, in conjunction with patient data.

In the following paragraphs, a series of methods and software tools designed to support these types of visualization will be presented.

**Clinical Algorithm Maps**

The most common way to represent clinical guidelines are the so-called "*flowchart algorithms*", also known as"*clinical algorithm maps*" [74], which are both quite known by doctors and eas y to use. One of the most important issues regarding the representation of clinical guidelines as "*clinical algorithm maps*" is the difficulty to represent complex situations, for which the space needed would be huge and an overview of the flowchart would be hard to follow. Temporal information is present only in a very basic way, depending on the positioning of a section (before, after). Moreover, they cannot be used to represent concurrent tasks and complex conditions such as those used in *Asbru* [74].

***Nassi-Shneiderman diagrams, PERT charts, Gantt charts, Petri Nets, and State Transition Diagrams***

All these are other ways to represent and visualize medical guideline sequences. Many of them are more expressive ways than "flowcharts", but none of them is complex enough (hierarchical decomposition, flexibility in the order of execution, status characteristics of guideline) to represent medical guidelines properly. None of them are accepted as a standard for the representation of medical knowledge [74].

***Protégé***

This is an open-source application for ontology development and media for acquiring information. The application is developed in *Java* and provides an extensible architecture for creating customized knowledge bases. *Protégé* (Fig. 2.8) allows the creation of medical guidelines in various models (e.g. *EON*, *Glif*, *Prodigy*, *PROforma*).

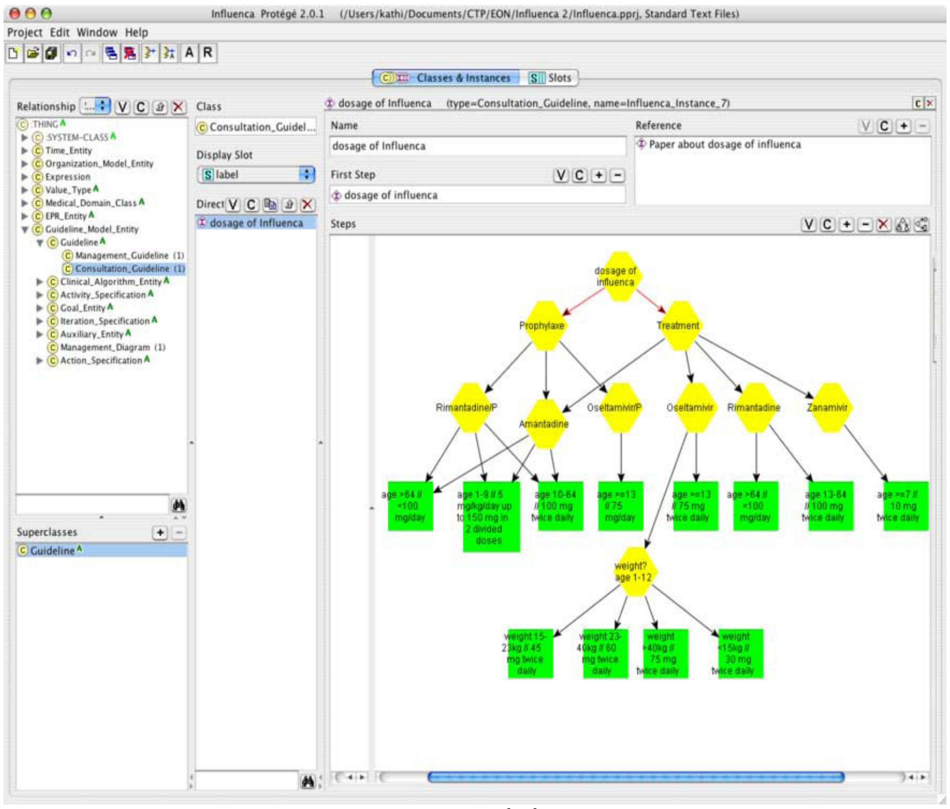


Fig. 2.8. *Protégé* [74]

### *VisiGuide*

It is a multi-ontology browser for guidelines, part of DeGeL. Its purpose is to present a number of guidelines grouped by semantic clues and to allow the user to view a guideline and explore its parts [74].

### *AsbruView – SopoView*

It is a graphical user interface developed in the project Asgaard / Asbru for authoring guidelines and protocols. Asbru is a complex language that cannot be understood very easily by doctors who have little knowledge about formalisms. AsbruView is a tool through which physicians can interact with Asbru environment, providing a graphical view of hierarchies and other concepts specific to medical guidelines. At this point, AsbruView provides four types of views:

- Topological View: displays the relations between guides in the form of plans without a specific time representation;
- Temporal View: focuses on issues of plans and conditions;
- *SOPOView*: offers a different view of temporal aspects of plans;
- *XML* View.

These tools for graphical representation have proven to be effective in communicating to physicians the concepts used in the plan-specification representation language *Asbru* (i.e. preferences, conditions, intentions) (Fig. 2.9) [74].

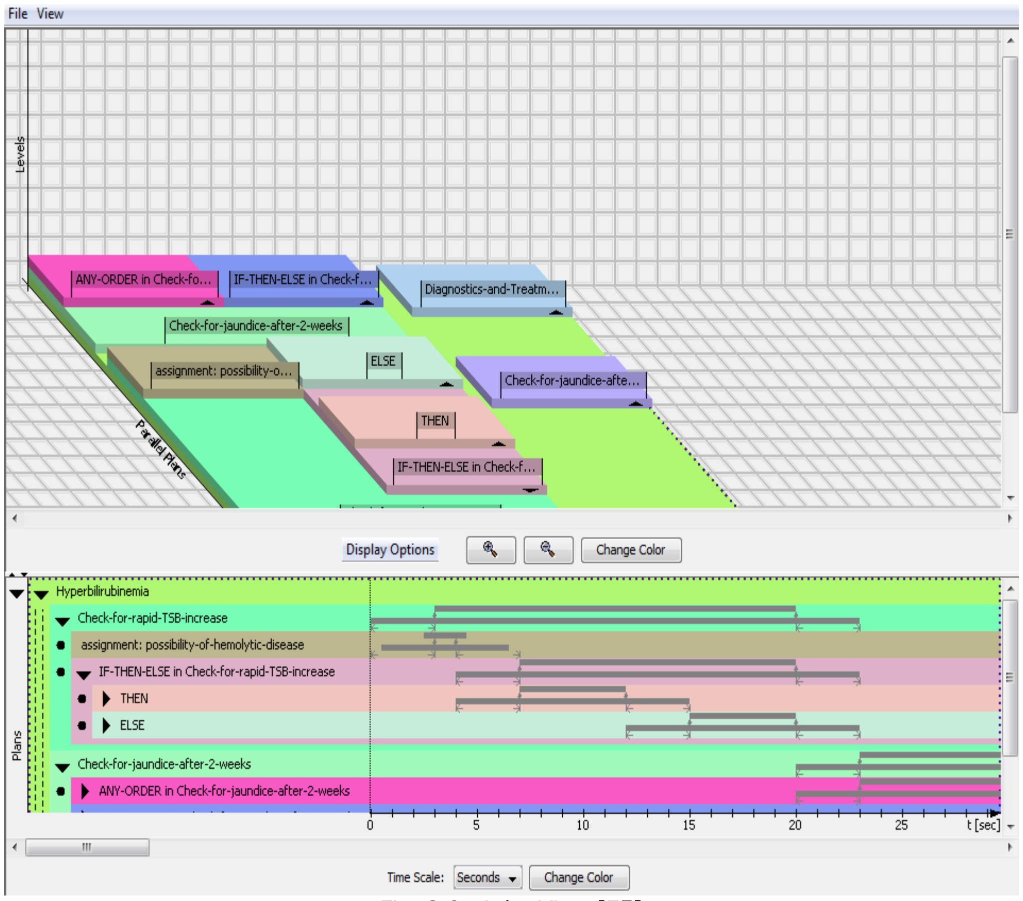

File View
Levels
ANY-ORDER in Check-fo...
IF-THEN-ELSE in Check-f...
Diagnostics-and-Treatm...
Check-for-jaundice-after-2-weeks
assignment: possibility-o...
ELSE
Check-for-jaundice-afte...
THEN
IF-THEN-ELSE in Check-f...
Parallel Plans
Display Options
Change Color
Hyperbilirubinemia
Check-for-rapid-TSB-increase
assignment: possibility-of-hemolytic-disease
IF-THEN-ELSE in Check-for-rapid-TSB-increase
THEN
ELSE
Check-for-jaundice-after-2-weeks
ANY-ORDER in Check-for-jaundice-after-2-weeks
Plans
0
5
10
15
20
25
t [sec]
Time Scale:
Seconds
Change Color

Fig. 2.9. *Asbru*View [75]

### *AsbruFlow*

It is a part of the prototype CareVis (Fig. 2.10) which helps to communicate the content and logic of Asbru treatment plans to doctors. AsbruFlow is based on the idea of "flowchart" as "clinical algorithm maps" which are well known among physicians. This concept has been extended to enable the description of the treatment plans in Asbru. It allows browsing inside guidelines, as well as detailed visualization of each component or specific area of the guideline.

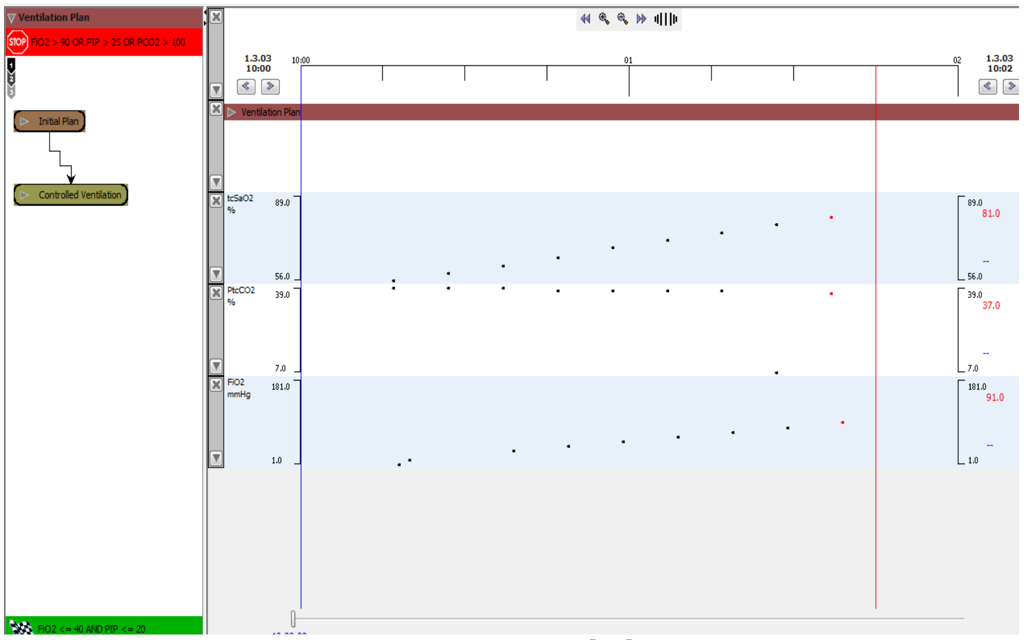


Fig. 2.10. *CareVis* [75]

In addition to these methods and applications described above, there are many other specialized tools for displaying computer-based guidelines. Several of them are: *Arezzo – Tallis Toolset, GLARE, Gravi++, VIE-VISU, Midgaard, DELT/A, GEM Cutter* [74].

## 2.5 Interfaces evaluation and directions to be followed

Interfaces are the user end part of a computer-based guideline approach. Here, the medical staff can see the steps of a protocol. The ways in which the steps are represented differ in various approaches.

After carrying out my own study, by observing the different ways of representing the protocol in execution, and after discussions with medical school students, five important features that interfaces should possess emerged [76], only some of them being implemented in the approaches discussed above. These five features are:

- The possibility to view data concerning the patient (for future needs we noted this feature as "Pvi") on the same interface with the protocol steps. Data concerning the patient may refer to data collected in real time from the patient, data representing the evolution of the patient, etc. The students said that they would prefer the possibility to choose from a list the data needed and the availability of a predefined list of data representative for the running protocol.

- The "alarm" ("Afi"). In an emergency situation an alarm will appear. The emergency situations can have different degrees of importance depending on the hospital departments where the protocol is implemented (e.g., an alarm is more useful in an intensive-care unit than in an orthopedic department).
- Treatment warning ("Twi"). This feature is similar to the "alarm", but a warning message will appear if the medical staff is about to apply an inadequate treatment for a certain patient.
- Multiple choices ("Mci"). The medical staff can choose between several suggested alternatives the treatment that he/she consider being the most appropriate. All of those choices should have a short description (e.g., who recommended the treatment, the success rate, a score).
- Feedback ("Fei".). This feature allows the medical staff to give feedback concerning the protocol (inadequate recommendations, user interface issues, suggestions for new interface features, etc.).

Considering these feature, we proposed an evaluation of the interfaces of the three approaches presented in the previous section. The respondents to our survey (students S1-S6) where inquired about the "importance" of each of the five features that interfaces should possess. The "importance" was in the range from 0 to 5, 0 signifying a feature of little and 5 for features of most importance [77]. The ratings given by the respondents are summarized in Table 2.3.

| | S1 | S2 | S3 | S4 | S5 | S6 | A |
|---|---|---|---|---|---|---|---|
| Pvi | 4 | 4.5 | 3.5 | 5 | 5 | 5 | 4.5 |
| Afi | 5 | 4 | 4.5 | 3 | 5 | 3 | 4.1 |
| Twi | 4 | 4 | 4 | 3.5 | 4 | 2 | 3.6 |
| Mci | 4 | 4 | 4 | 2.5 | 3 | 4 | 3.6 |
| Fei | 5 | 4 | 5 | 4 | 5 | 5 | 4.7 |

Table 2.3 Results of the survey regarding the "importance" of Pvi, Afi, Twi, Mci, Fei interface features

Based on the results of the survey and inspired by [78] we proposed the next formula for evaluating the interface:

$$Ev5=(0 \text{ or } 1)*A_{Pvi}+(0 \text{ or } 1)*A_{Afi}+(0 \text{ or } 1)*A_{Twi}+(0 \text{ or } 1)*A_{Mci}+(0 \text{ or } 1)*A_{Fei}+30, \quad (2.1)$$

Ev5 is the evaluation of the approach interface based on the five features.

$A_{Pvi}$, $A_{Afi}$, $A_{Twi}$, $A_{Mci}$ and $A_{Fei}$ represent the average "importance" of the features.

(0 or 1) – is 1 if the approach interface has this feature and 0 if the approach does not have it.

30 is the value that students sad that should be add for visualize only the protocol.

The score obtained by applying the proposed formula for *CareVis* (*Asbru*):

Ev=1*4.5+1*4.1+1*3.6+1*3.6+0*4.7+30=45.8

The score obtained by applying the proposed formula for *Tallis* (*PROforma*):

Ev=1*4.5+1*4.1+0*3.6+1*3.6+0*4.7+30=42.2

The score obtained by applying the proposed formula for *GLIF*:

Ev=0*4.5+0*4.1+0*3.6+1*3.6+0*4.7+30=33.6

Based on the scores obtained by applying the formula (1.1), it can be noted the compliance of medical school students with interfaces for representing medical protocols and also the complexity of these interfaces. The results show that *CareVis* is more complex and would be better accepted by the future physicians. It can also be noted that a feedback box and the possibility of viewing data concerning the patient represent the highest rated features; a well-accepted interface should contain these two features.

The evaluation of the interfaces can help the development of a new interface for a clinical decision support system which implements the concepts presented above, in order to increase the acceptance of *CDS*. This interface will be used for the visualization of medical recommendations and advices which are generated based on the solutions that will be presented in the thesis.

## 2.6 Conclusions

In this Chapter, an analysis of the clinical decision support domain has been made using 44 references. There were presented:

- the reasons for using narrative medical guidelines and protocols;
- the advantages of the computer-based medical guidelines and protocols;
- the main components of a clinical decision support system;
- the most important existing solutions in the domain;
- the evolution of the various approaches;
- a comparison between different approaches;
- the standards used for the representation of patient data;
- the tools used for the visualization of medical recommendations.

This analysis revealed the required improvements of clinical decision support in order to have more complete and precise medical recommendations:

**A. Increase of the interoperability of *CDS* by implementing standards for the representation of patient data and medical rules**

To enable the communication between different medical information systems, several standards were proposed: *HL7 CDA*, *vMR*. In order to reduce the gap between different medical knowledge representations (i.e. medical rules in: *ASTI*, *Asbru*, *PROforma*, GASTON etc.) and to allow the access of *CDSS*s to multiple medical data sources (e.g., patient data), an architecture which uses multiple *HL7 CDA*s as patient data source might improve the adaptability of *CDS* to local context.

Another solution would be the use of the *HL7 vMR* standard in conjunction with *HL7 CDA* documents, in order to make the *CDSS*s more adaptable by allowing access to more medical data sources used for the inference. Therefore, I proposed, developed and implemented an architecture which uses as inputs multiple *HL7 CDA* documents and data in *vMR* format.

Based on the analysis of features implemented by existing clinical decision support systems, I observed that *Arden Syntax* is one of the most used standard formalisms. It allows the rule-based representation of medical rules, which is easier to be understood by physicians. There are available open-source software applications which implement this standard (*Egadss, Arden2ByteCode*) that could be further developed. These are the reasons why the proposed architecture implements the *Arden Syntax* standard for the representation of medical rules.

**B. Increase of the medical knowledge base by using knowledge discovery methods**

In order to increase the medical knowledge base, I proposed a method to translate medical rules obtained by *data mining* techniques into *Arden Syntax* formalism. This translation is needed in order to use these new medical rules in different *CDSSs* which implement this standard.

Briefly, the contributions claimed by the author regarding the work presented in this Chapter are:

- The assessment of the present level of knowledge and development achieved in the field of clinical decision support, underlining its importance in improving the medical care act.
- The assessment of several of the most important existing approaches related to the generation of medical recommendations in terms of the two main modules of clinical decision support systems (authoring tools, execution engine).
- The assessment of the evolution of different *CDS* approaches in computer-based guidelines and protocols, starting from 1975 to 2012, and the improvement of an existing time-oriented graphical representation of the history of evolution of *CDSSs.*
- The comparison between the *CDS* solutions having the greatest impact in the literature based on a set of features which should be present in clinical decision support systems in order to be suited for as many situations as possible and to improve the acceptance of these systems.

The assessment of the interfaces of existing computer-based guideline solutions by using an original questionnaire-based approach and the development of a method for the evaluation of the user interface quality.

# 3. ACQUIRING PATIENT SPECIFIC DATA BY USING *HL7* STANDARDS

The use of standards for the representation of this data as inputs for *CDS* can increase the accuracy of medical recommendation by allowing the system to have access to multiple medical information systems which implement these standards. In order to generate medical recommendations medical rules are needed beside the patient data. To have access to as many medical rules as possible, the use of standards for their representation can increase the knowledge base of the system by allowing the import of computer representation of medical rules, eliminating the need of formalization of certain narrative guidelines (which were already formalized).

In this Chapter, a solution for the acquisition of patient specific information from multiple sources using different *HL7* standards (*CDA, vMR*) is presented. Beside the use of standards for the representation of patient data, there are also used *HL7* standards for the representation of medical rules, in order to increase the *CDS* interoperability.

## 3.1. Standards necessity

Although clinical decision support systems have great potential to improve the quality of patient care and safety, only a few reliable capabilities of *CDS,* beside the one related to drug prescriptions, are widely available. The limited *CDS* implementation in different health care facilities is due to the lack of standard models for representation of clinical information and associated terminologies (used within health care institutions and health information systems) and *CDS* resources [41].

To enable the communication between different medical units, some standards can be applied: *HL7 CDA, vMR HL7, Arden Syntax* [79, 80]. Consequently, the medical decision system has to implement the same standard as the one chosen for communication. Implementing the *HL7* standards for retrieving data improves the access to information from different institutions where the different components of the system are installed. In this manner, the recommendations which are generated are more precise, based on a more complete set of data.

## 3.2. Clinical decision support system architecture

In this section, is presented a system architecture which I developed for the generation of medical recommendation, based on the use of multiple patient data sources and implementing different *HL7* standards for the representation of patient data and medical knowledge [81].

The main components of the system architecture are (Fig. 3.1):

- Data Manager;

- HL7 CDA Component;
- Egadss - inference engine;
- TM-vMR;
- Interface.

In the first part of this section, the presentation will be focused on the use of the *HL7 CDA* standard in collaboration with the *Data Manager* and inference engine.

In the second part, the presentation will be focused on the use of the *vMR* standard in conjunction with the *Topic Maps* technology and the way in which this can be integrated with the existing solution.

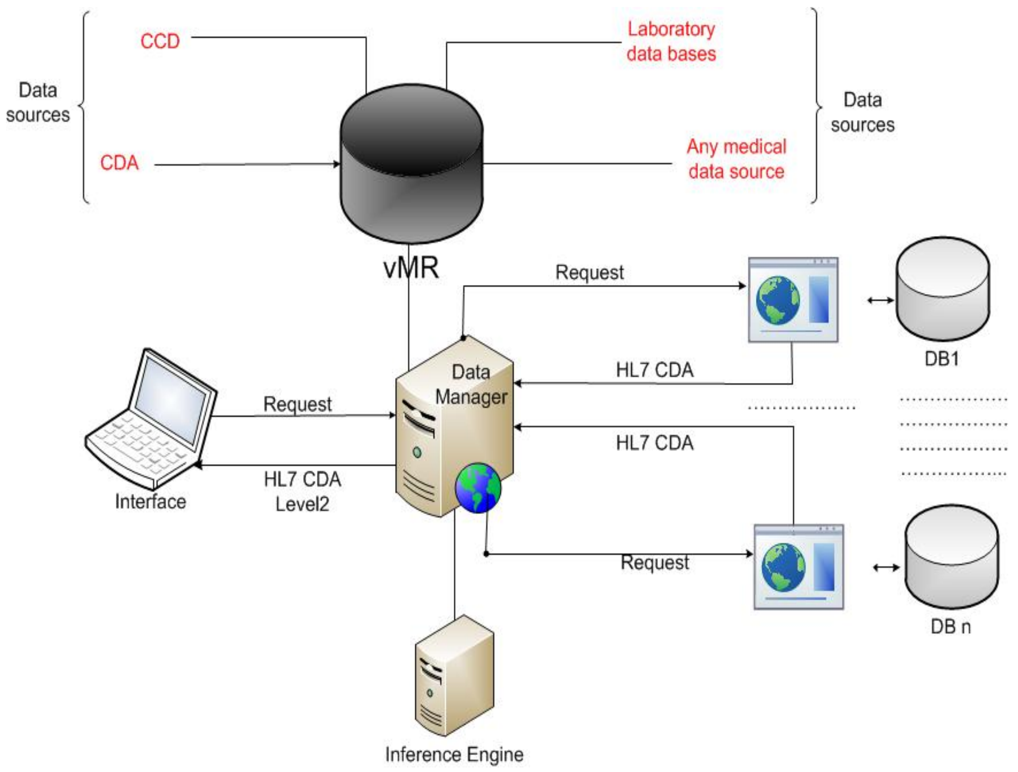


Fig. 3.1. Clinical decision support system architecture: *Interface,* inference engine, *Data Manager*, *TM-vMR*, *HL7 CDA Component*

A short description of the main components of the systems is realized in the next paragraphs.

***HL7 CDA Component***

The *HL7 CDA Component* has been developed in order to obtain information from different databases (e.g., a laboratory or radiology database) and then represent it in *HL7 CDA* format and send it to the decision system. The *HL7 CDA Component* implementation is made in *Visual Studio.Net 2008*, in C# language (as a web service). The databases for the current activity are implemented on *SQL Server 2008*, but the solution is similar for *Oracle* or *MySQL*.

***Egadss*, Inference Engine**

The inference engine is based on *Egadss* open-source solution [16, 21], which allows as inputs *HL7 CDA* (Level 3) standard messages (as *XML* files built from the individual *XML* files retrieved from the each *HL7 CDA Component*.), containing patient data, in order to have a standardized communication interface between databases and "recommendation generator". Another standard used by *Egadss* is *Arden Syntax*, which is a clinical guideline formalism accepted as an official standard by *HL7* group for the representation of medical rules. The result of the inference is a *CDA Level 2* document, containing the medical recommendations [16, 21].

***Data Manager***

The communication between all the components of the *CDS* is based on web services, representing: *HL7 CDA Component*s, decision making system or the interface. To manage the connection and the order in which the different web services are called, a *Data Manager* was developed. *Data Manager* has the role to respond at different requests that come from the main components of the system (interface, medical data source, inference engine). In order to realize this, three communication channels are opened with: Interface, *HL7 CDA Component* and the inference engine. Beside the use of *HL7 CDA* documents other sources can be added to the system through the *Data Manager*.

***TM-vMR***

One of these sources can be a *virtual Medical Record (vMR)* that implements the specifications from [*11*, 28]. The *vMR* is a standard proposed by *HL7* group for the representation of patient data for the clinical decision support systems. The *vMR* is implemented with *Topic Maps* technology.

***Interface***

The interface allows the medical staff and the patients to visualize the steps of the protocols, medical information regarding a patient, different alarms, as well as inserting feedback concerning the recommendations. The interface is implemented using *ASP.Net* platform with C#. A more detailed description of the architecture can be found in [81].

## 3.3. Architecture components based on the *HL7 CDA* standard and *Arden Syntax*

In this section is presented the part of the system which is based on the use of *HL7 CDA* standard as patient data sources (Fig. 3.2).

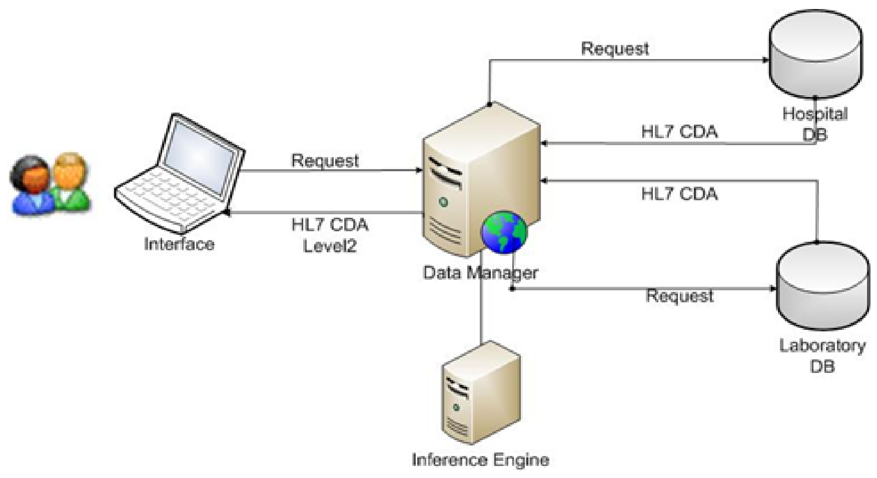


Fig. 3.2. *HL7 CDA* standard based architecture

### 3.3.1. *HL7 CDA Component*

For the current application the *HL7* version 3 messages are presented as *HL7 CDA* messages created by the *CDA Component*. The *Sending HL7 Application* is the *CDA* Component which extracts the data needed and converts it into a *CDA* in *XML* format, then sends it to the *Receiving HL7 Application*. The *Messaging Infrastructure Layer* is responsible for the *HL7* messages transfer, following the rules specified by the *HL7* applications. For the presented process the Message Transport Layer is built on TCP/IP, the communication protocol for the Internet, and defines the rules for computer communication on Internet [44]**.**

The communication between the *Data Manager* and automatic generator is done by using *HL7 CDA*. The *CDA Component* takes the needed information from database, then converts it into an *XML HL7 CDA* file and sends the information over the *SOAP* protocol to the automatic generator.

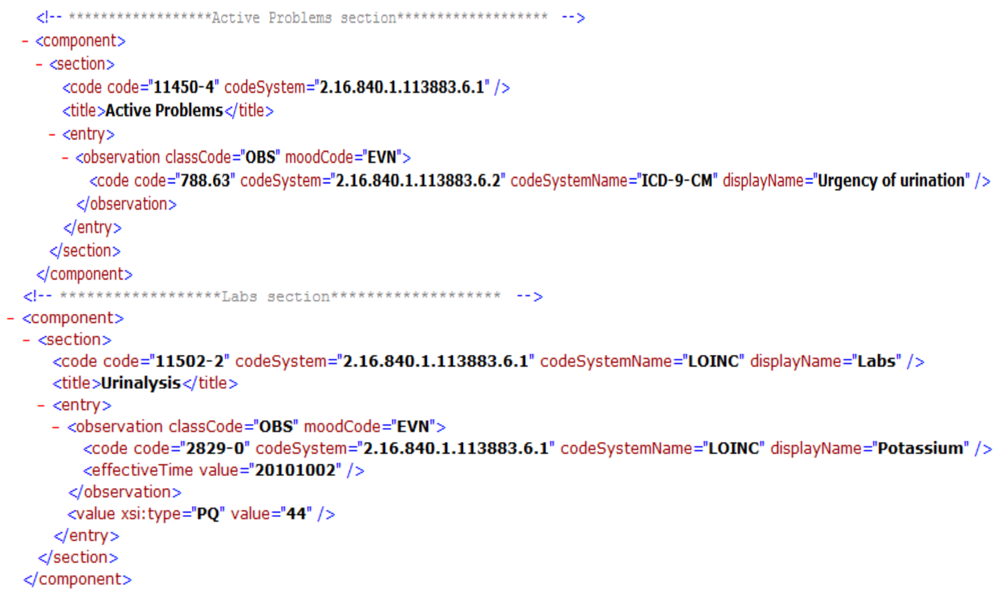
#### *Component description and implementation*

The *HL7 CDA Component* extracts the data from a local database and presents it as *HL7 CDA* in *XML* format and provides it to inference engine (Vida et al [82]).

The *HL7 CDA Component* implementation is made in *Visual Studio.Net 2008*, in C# language. The databases for the current activity are on *SQL Server 2008*, but the solution is the same for *Oracle* or *MySQL*. The *HL7 CDA* message formatting is implemented currently in C#, but it can be implemented also in *Java*, the only requirement being the *XML* format (Vida et al [82]).

The *HL7* component represents a major improvement of the existing solutions, allowing the decision making part to have access to more complex information, besides the one that can be found only in local databases.

An example of an *XML* file is presented in Fig. 3.3, containing the data concerning the laboratory results in *HL7 CDA* format. In the *Laboratory* section the value of potassium is represented and for this an *ICD-9* (*International Classification of Diseases*) code containing the corresponding diagnosis is associated [83]. This information is used by the inference engine.



```
<!-- *****************Active Problems section****************** -->
- <component>
  - <section>
      <code code="11450-4" codeSystem="2.16.840.1.113883.6.1" />
      <title>Active Problems</title>
    - <entry>
      - <observation classCode="OBS" moodCode="EVN">
          <code code="788.63" codeSystem="2.16.840.1.113883.6.2" codeSystemName="ICD-9-CM" displayName="Urgency of urination" />
        </observation>
      </entry>
    </section>
  </component>
<!-- ******************Labs section****************** -->
- <component>
  - <section>
      <code code="11502-2" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC" displayName="Labs" />
      <title>Urinalysis</title>
  - <entry>
    - <observation classCode="OBS" moodCode="EVN">
        <code code="2829-0" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC" displayName="Potassium" />
        <effectiveTime value="20101002" />
      </observation>
      <value xsi:type="PQ" value="44" />
    </entry>
  </section>
</component>
```

Fig. 3.3. Example of *HL7 CDA* message

### 3.3.2. *Egadss*, inference engine

#### 3.3.2.1. *Egadss*

*Egadss* is a clinical decision support system that has as inputs *HL7 CDA* standard messages as *XML* files [21, 16]. This results in a standardized communication interface between databases and "recommendation generator". *Egadss* extracts the patient data from the *XML* and uses this information to generate new clinical recommendations, making the system very flexible and easy to be implemented in any medical unit. The recommendations resulted from *Egadss* are also structured as a *CDA Level 2* document.

Another advantage of *Egadss* is the use of *Arden Syntax* for the representation of the guidelines. *Arden Syntax* has as main characteristics:

- accepted as an official standard by *HL7* [23];
- is a textual language;
- easy understood by physicians;
- freely available.

These are the reasons why *Arden Syntax* is cohoused as guideline formalisms instead of *PROforma*, *GLIF* (*GELLO*), *Asbru*, etc.[ 1, 16, 21, 84].

*Egadss* is an already developed system and extra value is added by using multiple data sources based on *HL7* standards, as in the described application [85].

### 3.3.2.2. *Egadss* functionality

Based on the patient data which is received through the *Data Manager*, *Egadss* has access to data from different clinical institutions offering more precise medical recommendations based on this information.

In order to obtain medical recommendations based on the data received from the *Data Manager*, *Egadss* uses medical rules represented with *Arden Syntax*, as a knowledge source for its expert system [21]. Currently *Egadss* uses *ICD9*, and in Romania is used *ICD*10 [86].

In order to exemplify the functionality, *Egadss* was tested for the management of chronic pelvic pain (chosen as the model guideline used to test the data) adapted to the Romanian situation. The "*Management of chronic pelvic pain*" is presented in Fig. 3.4 [87]. I wrote a *MLM* for the branch of the protocol that represents the recommendation that should be made in case that the patient had "voiding symptoms" and he made a urinalysis (the right branch from Fig. 3.4). In Fig. 3.5, the *Knowledge* part of the *MLM* (*Medical Logic Module*) used to represent this part of the guideline can be seen. A presentation of the *Arden Syntax* can be found in section 2.2.1.

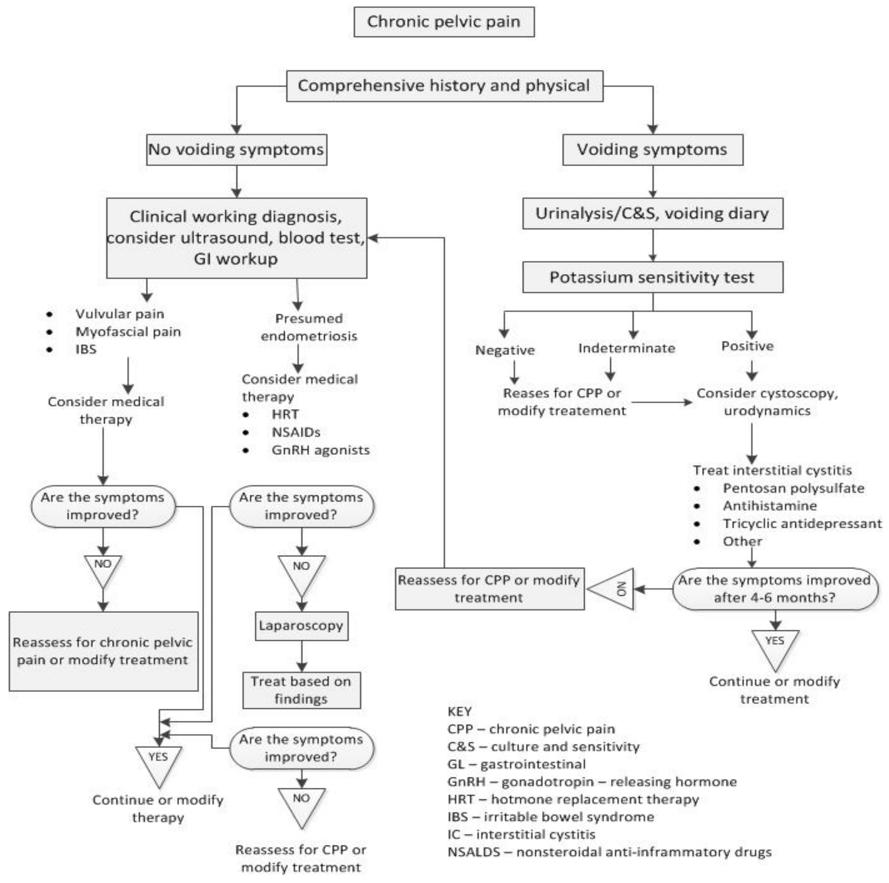
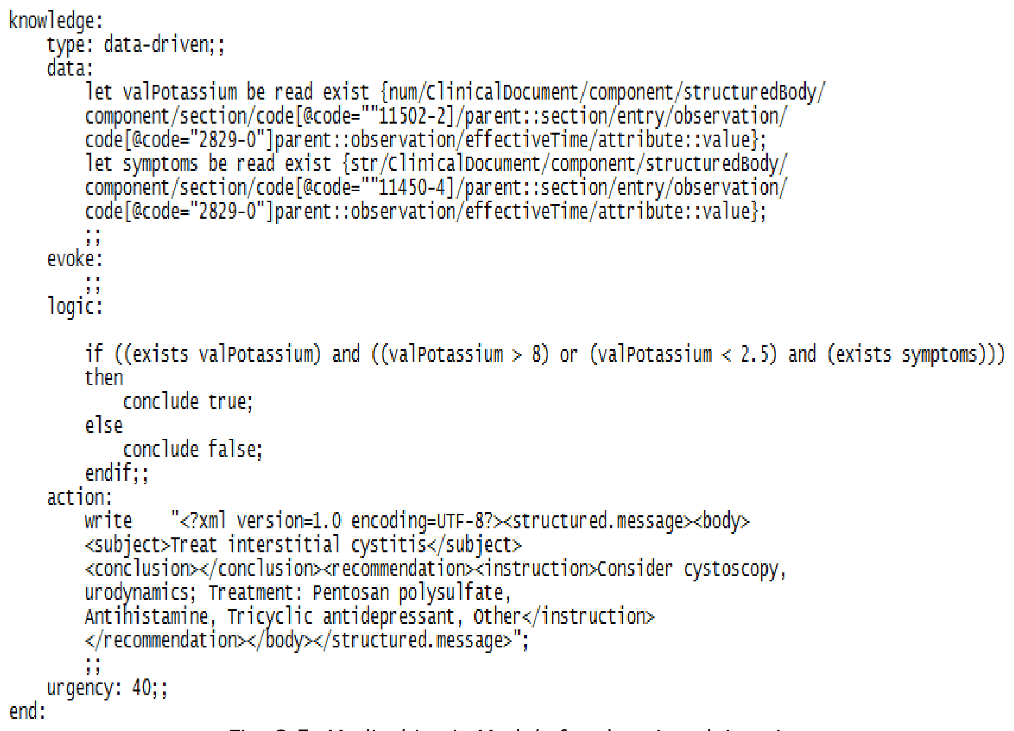

Chronic pelvic pain
Comprehensive history and physical
No voiding symptoms
Voiding symptoms
Clinical working diagnosis, consider ultrasound, blood test, GI workup
Urinalysis/C&S, voiding diary
Potassium sensitivity test
Vulvular pain
Myofascial pain
IBS
Presumed endometriosis
Negative
Indeterminate
Positive
Consider medical therapy
Consider medical therapy
HRT
NSAIDs
GnRH agonists
Reases for CPP or modify treatement
Consider cystoscopy, urodynamics
Treat interstitial cystitis
Pentosan polysulfate
Antihistamine
Tricyclic antidepressant
Other
Are the symptoms improved?
Are the symptoms improved?
NO
NO
Reassess for CPP or modify treatment
NO
Are the symptoms improved after 4-6 months?
Laparoscopy
Reassess for chronic pelvic pain or modify treatment
YES
Treat based on findings
Continue or modify treatment
Are the symptoms improved?
YES
Continue or modify therapy
NO
Reassess for CPP or modify treatment
KEY
CPP – chronic pelvic pain
C&S – culture and sensitivity
GL – gastrointestinal
GnRH – gonadotropin – releasing hormone
HRT – hotmone replacement therapy
IBS – irritable bowel syndrome
IC – interstitial cystitis
NSALDS – nonsteroidal anti-inframmatory drugs

Fig. 3.4. Management of chronic pelvic pain [87]

The next step in the generation of medical recommendation is to obtain the necessary medical rules, based on the *MLM* file read by *Egadss*, which translates the medical rules from the *MLM* into *CLIPS* rules. These are then used by the expert system from *Egadss* to generate the medical recommendations. *CLIPS* is the core for the guideline execution mechanism in *Egadss* [16].

After running the *CLIPS* rules on the patient data, new recommendations were generated. The patient data are extracted from the *XML* (section 1.2) received from the *Data Manager*.



```
knowledge:
    type: data-driven;;
    data:
        let valPotassium be read exist {num/ClinicalDocument/component/structuredBody/
        component/section/code[@code=""11502-2]/parent::section/entry/observation/
        code[@code="2829-0"]parent::observation/effectiveTime/attribute::value};
        let symptoms be read exist {str/ClinicalDocument/component/structuredBody/
        component/section/code[@code=""11450-4]/parent::section/entry/observation/
        code[@code="2829-0"]parent::observation/effectiveTime/attribute::value};
        ;;
    evoke:
        ;;
    logic:

        if ((exists valPotassium) and ((valPotassium > 8) or (valPotassium < 2.5) and (exists symptoms)))
        then
            conclude true;
        else
            conclude false;
        endif;;
    action:
        write    "<?xml version=1.0 encoding=UTF-8?><structured.message><body>
        <subject>Treat interstitial cystitis</subject>
        <conclusion></conclusion><recommendation><instruction>Consider cystoscopy,
        urodynamics; Treatment: Pentosan polysulfate,
        Antihistamine, Tricyclic antidepressant, Other</instruction>
        </recommendation></body></structured.message>";
        ;;
    urgency: 40;;
end:
```

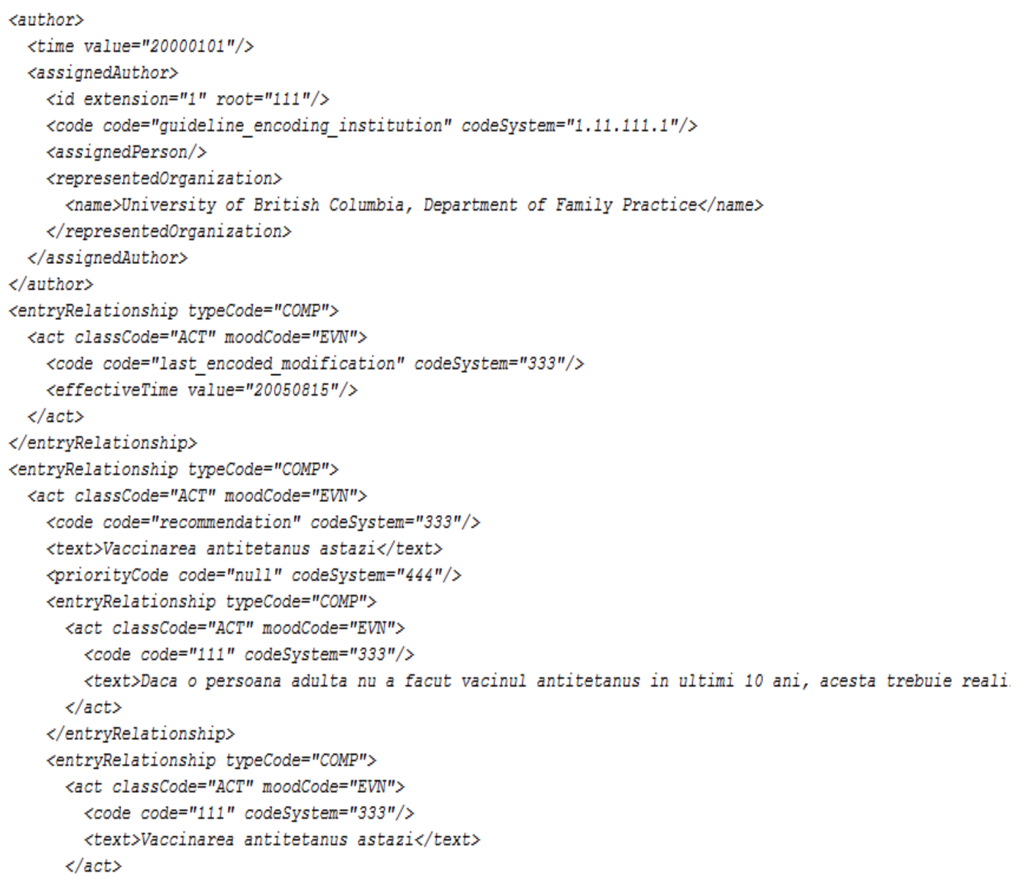
Fig. 3.5. *Medical Logic Module* for chronic pelvic pain

The following steps are driven during the execution of recommendations:

1. Compiling *MLM* and loading rules in *CLIPS*;
2. Awaiting the arrival of patient data;
3. Extracting data from *CDA* (using *XPath* which is a language used for the extraction of information from *XML*) based on the address specified in *MLM* and then load them as *CLIPS* facts;
4. Execution of *CLIPS* rules;
5. Export of results in *CDA Level 2* format (Fig. 3.6);
6. *CLIPS* reset for the next patient;
7. Returning to step 2.

### 3.3.2.3. Modification of *Egadss*

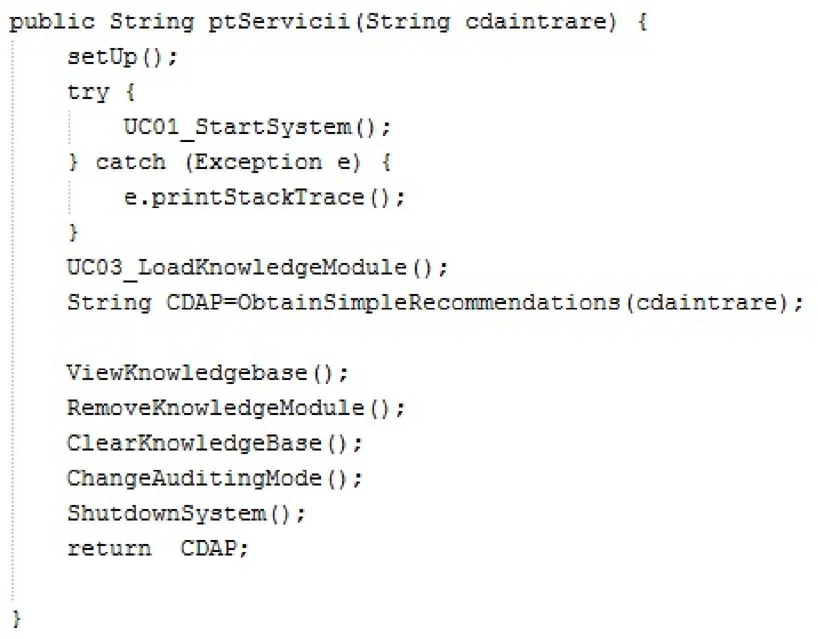
The communication between the inference engine (*Egadss*) with *HL7 CDA Component*, *TM-vMR* and the interface is performed through the *Data Manager*. To enable this communication was necessary to develop a special package which has as main component "*Egadss*GetRecommendations" web service. This web service can be called by the *Data Manager*. This service receives as input *HL7 CDA L3* documents (containing the patient data) which are then transmitted for the inference.



```
<author>
  <time value="20000101"/>
  <assignedAuthor>
    <id extension="1" root="111"/>
    <code code="guideline_encoding_institution" codeSystem="1.11.111.1"/>
    <assignedPerson/>
    <representedOrganization>
      <name>University of British Columbia, Department of Family Practice</name>
    </representedOrganization>
  </assignedAuthor>
</author>
<entryRelationship typeCode="COMP">
  <act classCode="ACT" moodCode="EVN">
    <code code="last_encoded_modification" codeSystem="333"/>
    <effectiveTime value="20050815"/>
  </act>
</entryRelationship>
<entryRelationship typeCode="COMP">
  <act classCode="ACT" moodCode="EVN">
    <code code="recommendation" codeSystem="333"/>
    <text>Vaccinarea antitetanus astazi</text>
    <priorityCode code="null" codeSystem="444"/>
    <entryRelationship typeCode="COMP">
      <act classCode="ACT" moodCode="EVN">
        <code code="111" codeSystem="333"/>
        <text>Daca o persoana adulta nu a facut vacinul antitetanus in ultimi 10 ani, acesta trebuie reali
      </act>
    </entryRelationship>
    <entryRelationship typeCode="COMP">
      <act classCode="ACT" moodCode="EVN">
        <code code="111" codeSystem="333"/>
        <text>Vaccinarea antitetanus astazi</text>
      </act>
```

Fig. 3.6. Medical recommendation represented in *HL7 CDA Level 2* format

From this web service a *Java* function (Fig. 3.7) is called in order to generate the medical recommendations. This function uses the already build function of *Egadss* open-source software.



```
public String ptServicii(String cdaintrare) {
    setUp();
    try {
        UC01_StartSystem();
    } catch (Exception e) {
        e.printStackTrace();
    }
    UC03_LoadKnowledgeModule();
    String CDAP=ObtainSimpleRecommendations(cdaintrare);

    ViewKnowledgebase();
    RemoveKnowledgeModule();
    ClearKnowledgeBase();
    ChangeAuditingMode();
    ShutdownSystem();
    return  CDAP;

}
```

Fig. 3.7. Java function for initiate the inference based on *Egadss* open source

This function has the role to:

- start the *Egadss* system;
- call the function which deals with the management of *XML* parsing and starting the inference;
- to call a set of functions in order to stop the inference engine.

This web service is implemented using *Metro* [88] technology (*Egadss* being developed in Java). Sun Microsystems developed the *Metro Web services stack*, which is open source. It incorporates the reference implementations of the *JAXB 2.x* data-binding and *JAX-WS 2.x* Web services standards and other *XML*-related Java standards [88].

### 3.3.3. *Data Manager*

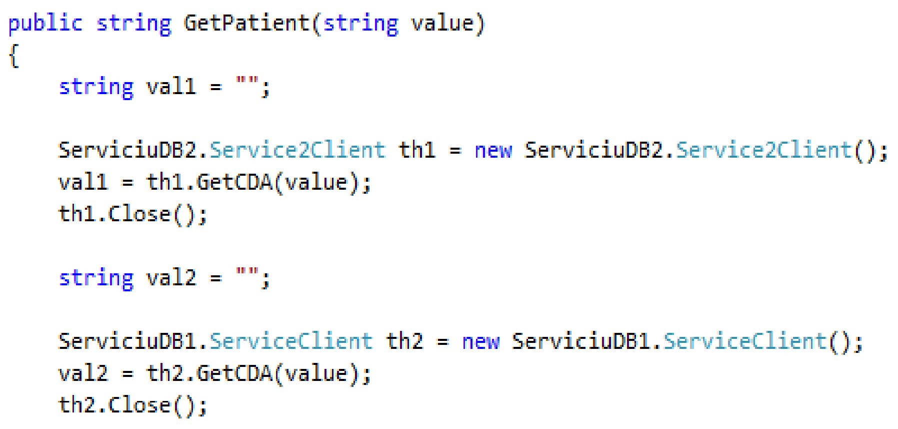
The *Data Manager* has the role to respond at different requests that come from the main components of the system (interface, medical data sources, inference engine. Based on the patient ID obtained from the interface, the *Data Manager* calls all the web services (*HL7 CDA Components* and *TM-vMR*) in order to retrieve data from different institutions. The web services get the information (as stated in section 3.1 and 3.4) and send to the manager the *HL7 CDA* documents and the data from the *vMR*. At this point the *Data Manager* has to merge all the information received and then pass this information to the inference engine. The inference engine reasons on the patient data and replies to the *Data Manager* the *HL7 CDA level 2* document containing the medical recommendations. This document is then passed to the interface.

To achieve the communication between the *Data Manager* and the *HL7 CDA Components* web services are implemented using *Windows Communication Foundation* [WIN 12]. In this manner, the client (*Data Manager*) can call all the *HL7 CDA Components (*services*)* located in different medical units, which have access to the Internet. In a similar way, the communication with the inference engine and *TM-vMR,* is achieved. For this, the *Data Manager* calls a web service that is implemented in *Java*. Using the web services, the inference engine can be placed in different locations.

The *Data Manager* is seen as a set of services by the *Interface*. Thus, the entire system of generating medical recommendations is initiated through the interface, which transmits to the *Data Manager* the patient ID for whom to generate the recommendation. This interface communicates with *Data Manager* through two services: "*getData*" and "*getRecommendations*".

***getData***

The service "*getData*" (Figure 3.8) receives as parameter the patient ID. The ID is used to call the *HL7 CDA Components* and the *TM-vMR*, which act as web services for the *Data Manager*. These services, after extracting the patient data, send it to the *Data Manager*. In this way, multiple *CDA Components* can be accessed (Figure 3.8).



```
public string GetPatient(string value)
{
    string val1 = "";

    ServiciuDB2.Service2Client th1 = new ServiciuDB2.Service2Client();
    val1 = th1.GetCDA(value);
    th1.Close();

    string val2 = "";

    ServiciuDB1.ServiceClient th2 = new ServiciuDB1.ServiceClient();
    val2 = th2.GetCDA(value);
    th2.Close();
```

Fig. 3.8. *getData* web services used to call two *CDA Components*

This service has also to return a unique *HL7 CDA document*. The service extracts the information from the multiple received documents and retains only the last value of a data type (if the same data type is present in multiple *HL7 CDA* documents or in *vMR* format).

***getRecommendations***

The service "*getRecommendations*" calls "*Egadss*GetRecommendation" (web services) which is implemented with *Metro* technology (in *Java*), being the connecting part between the *Data Manager* (*.Net* platform) and *Egadss* (*Java* platform). The "*EgadssGetRecommendation*" service call is made based on the patient data obtained by *"getData"*. "*EgadssGetRecommendation*" (service in *Egadss*) returns to the data manager recommendations resulting from the inference. The Clinical recommendations (for a patient) are represented in *HL7 CDA L2 format*. The recommendations are then sent to the interface.

The *Data Manager* can be accessed from outside through two services:

- "*GetData*" – which has as input the patient ID and has as output the patient's clinical data in *HL7 CDA* format;
- "*GetRecommendation*" - which has as input the patient ID and has as output the medical recommendations for the specified patient (*HL7 CDA L2* format).

In turn it calls the next web services:

- "*HL7 CDA components*" – for retrieving information from institutions which implement *HL7 CDA* standard;
- "*EgadssGetRecommendation*" – in order to call the inference engine;
- *"vMRData"* – which retrieves information from the a virtual medical record.

## 3.4 Using the *virtual Medical Record* standard and *Topic Maps* technology as patient data sources for *CDS*

Any knowledge representation technique has to be developed in such a way to fully cover the domain it deals with. The healthcare domain is very large, complex, and in continuous changing. Aiming to achieve better capabilities for their work, medical personnel and researchers from the healthcare and information technology domains often face the need of a data model able to represent this domain in a more accurate manner.

In order to solve this problem I developed and implemented a method to integrate the *HL7 vMR* standard (for the representation of patient data) in the previous architecture (Fig. 3.9).

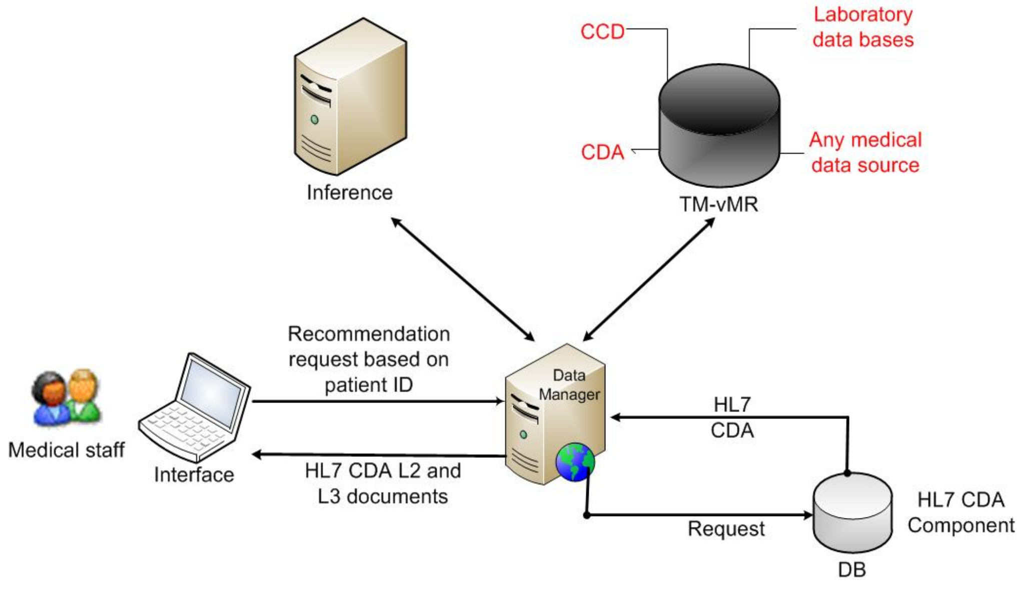
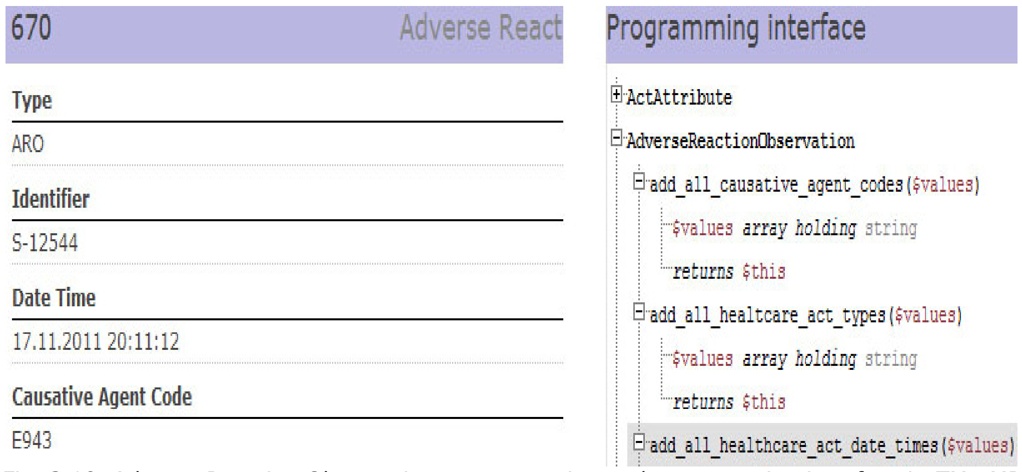


Fig. 3.9. Integration of *vMR* standard in the *Data Manager*

The first problem of representing and integrating the *vMR* standard with the existing solution was to find a way to represent the *vMR* data model. The solution for this was the use of *Topic Maps* technology. It is a new technology that is suited for the representation of data models, bringing a series of advantages (extensibility, easy representation of the connections between data), which are further described in the next section.

### 3.4.1. ***TM-vMR*** knowledge base

A definition for *Topic Maps* is: "*Topic Maps* is a technology for encoding knowledge and connecting this encoded knowledge to relevant information resources. *Topic Maps* are organized around topics, which represent subjects of discourse; associations, representing relationships between the subjects; and occurrences, which connect the subjects to pertinent information resources" [89]. Instances of *TM* standard are called *Topic Maps*, and they are digital representations of knowledge about different domains made in a subject-oriented manner [90, 91]. *Topic Maps* could be used to represent "anything whatsoever" [89]. That is why this technology is suited for the representation of the vast medical knowledge domain concerning *CDS* [92]. Using *Topic Maps* [93, 94] the data elements from the *vMR* are represented as the topics of a *topic map* and the relations between these elements are represented as associations.



670 Adverse React

Type
ARO

Identifier
S-12544

Date Time
17.11.2011 20:11:12

Causative Agent Code
E943

Programming interface

```
ActAttribute
AdverseReactionObservation
  add_all_causative_agent_codes($values)
    $values array holding string
    returns $this
  add_all_healtcare_act_types($values)
    $values array holding string
    returns $this
  add_all_healthcare_act_date_times($values)
```

Fig. 3.10. Adverse Reaction Observation representation and programming interface in *TM-vMR*

The software used to represent the *vMR* as *topic map* is *Topincs*, an easy-to-use development tool for the web databases, based on the *TM* standard [43], allowing an easy representation of *vMR* data elements ("topics and associations"). The software allows the *topic map* developer to customize the behavior of the application, through a programming interface (Fig. 3.10, right side). This programming interface is automatically generated based on the serialization names (names assigned to all relevant items in the data bases) corresponding to the elements within the *topic map* schema of the represented domain. To make the content of the *topic map* accessible, a *PHP* object is used. In *Topincs* this object is named *tobject*, and its virtual programming interface is driven by the constraints of its topic type [43]. Every medical concept presented in the *vMR-Domain Analysis Model* (e.g., *encounter, observation, substanceAdministration, Adverse Reaction Observation*) is accessed using a *tobject* (Fig. 3.10, left side). In order to add new elements to the knowledge base, we used the *Topincs* interface for the *vMR*. Different functions from the programming interface are called to interact with the *tobjects*. Separate functions for deletion, creation or modification of a *tobject*, which are not present in the interface, can also be called (Dragu et al [95]).

From the physicians' perspective, this application could be viewed as an easy-to-use knowledge base, with an emphasis on the information retrieval. An informational construct within this knowledge base, a topic, is a conceptualization of a subject, and it is presented to the user in terms of: description of the subject, links to resources relevant to the subject, relationships and/or perspectives in which the subject is involved, hierarchical related topics, and so on.

From the point of view of information technology applied in healthcare, this approach can be an easy-to-implement solution for the medical knowledge web databases, with reduced costs, based on well-known technologies (*Apache, MySQL, PHP – AMP* solution) (Dragu et al [95].

### 3.4.2. *CDS* connection to *TM-vMR* module

The connection between the existing system and the *TM-vMR* represented in *Topincs* open-source is done with web services. These services are consumed from the *Data Manager*. This is a service-oriented architecture where the web service provider is the *TM-vMR* and the consumer is the *Data Manager.*

*NuSOAP* is the technology used for the development of web services in *PHP* for the access to the *TM-vMR* ontology . This is defined as "a group of *PHP* classes that allow developers to create and consume *SOAP* web services" [96].

These services allow the work with the patient data that is represented in the *TM-vMR* ontology through topic map objects (in "*Topincs*" are "*tobjects*"). The *tobjects* allow the insertion, deletion, modification and many other types of special functions to work with patient data from the *TM-vMR* (Fig. 3.11). The web services allow the connection of the model with the existing *CDS* through the *Data Manager*. Based on the data needed for a certain patient, the *Data Manager* calls the web services from the *TM-vMR* knowledge base, in order to generate new medical recommendations.

```
$obiectul = Tobject::get('LaboratoryObservationCode');
$val = $obiectul->get_value();
return $val
```

Fig. 3.11. Retrieving laboratory results based on *tobjects*

To connect the *Data Manager* to the *TM-vMR,* the *Service Model Metadata Utility Tool* is used, generating the needed proxy (set of functions) to enable the communication between *PHP* web services (*TM-vMR*) and *.Net* client (*Data Manager*).

The obtained proxy is then used in the *Data Manager* by the "*getData*" web services in order to call the *PHP* web services and access the patient information from the *TM-vMR*. *GetData* puts the data about the patient in the format accepted by *Egadss*, as described for the *HL7 Components* in section 3.3.3.

My system has as input an already build *topic map*. In this manner, the validity of the contained data depends on the manner in which the *Topic Maps* was build. In the building of *Topic Maps,* the subjectivism of the links between medical data cannot be removed because of the human factor involved in its development. My work regards only the presentation of a methodology to use a *vMR* represented with the help of *Topic Maps* as patient information source in clinical decision support systems. It is not the intention of this research to demonstrate the validity of the information represented in the topic map. It depends on the integration of the patient data and the validity of the links between this information based on medical experts contribution.

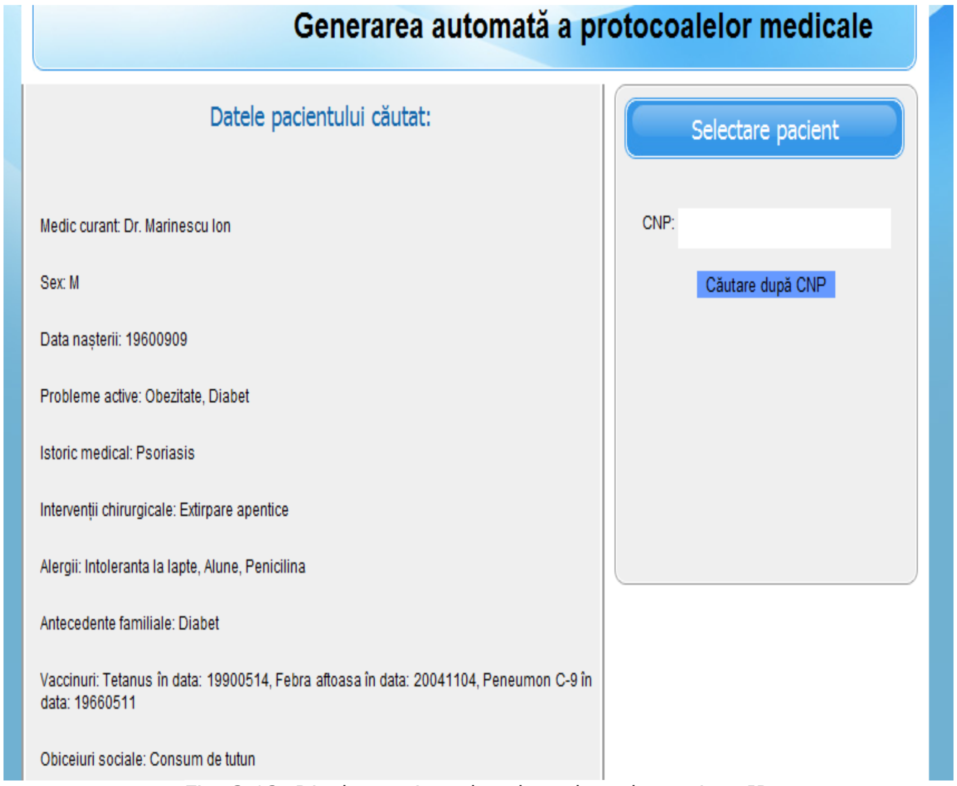
## 3.5. The visualization interface

Interfaces are the user end part of a computer-based guideline approach. Here, the medical staff can see the steps of a protocol. The way the steps are represented differ through various approaches. Methods and software developed to support recommendation visualizations are: *Protege, VisiGuide, AsbruView, GOT, Arezzo, GLARE, CareVis, AsbruFlow* [2, 16, 74, 97, 98]. Analyzing and comparing different existing interfaces in the field can help the design of a new better interface solution. This solution has to implement different well appreciated features (by medical staff, presented in other papers [74, 76]) of the existing interfaces, or to implement characteristics which were omitted in those but which are asked for by the medical staff.

Based on the evaluation of the interfaces from section 2.8 several characteristics emerged as necessary for the development of an interface:

- The possibility to view data concerning the patient on the same interface with the protocol steps;
- The "alarm" – in an emergency situation an alarm will appear;
- Treatment warning;
- Medical feedback;
- Multiple choices for medical staff.

The users are allowed to view data belonging to certain patients and medical recommendations based on the patient data. Other features of the interface that should be implemented are: different alarms, feedback offered by the medical personal regarding the correctness of the recommendations or some other problems regarding features of the *CDSS*. The user interface is actually a web application that was developed using the *ASP.Net* platform with C#.


Generarea automată a protocoalelor medicale

Datele pacientului căutat:

Medic curant: Dr. Marinescu Ion

Sex: M

Data nașterii: 19600909

Probleme active: Obezitate, Diabet

Istoric medical: Psoriasis

Intervenții chirurgicale: Extirpare apentice

Alergii: Intoleranta la lapte, Alune, Penicilina

Antecedente familiale: Diabet

Vaccinuri: Tetanus în data: 19900514, Febra aftoasa în data: 20041104, Peneumon C-9 în data: 19660511

Obiceiuri sociale: Consum de tutun

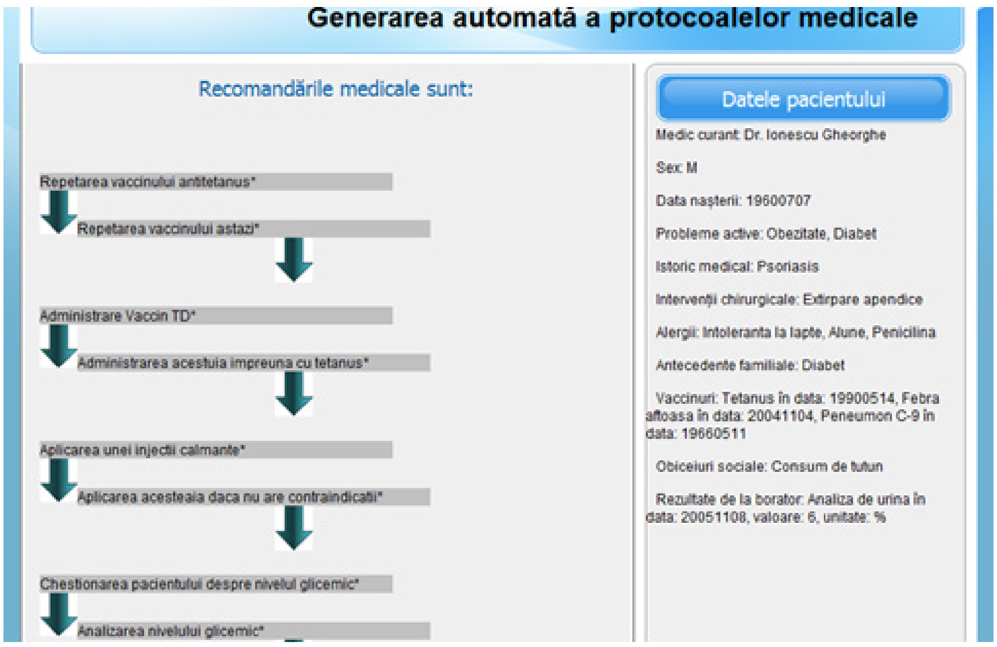
Fig. 3.12. Display patient data based on the patient ID

Getting to the website of the system, the users can find a webpage where are described the computer-based guidelines. To access the other pages of the application one needs to be logged in.

Another component ("*Search*") is a section that allows the visualization of a certain patient's data. Here, through the *Data Manager* and based on the patient ID a request for the patient data is made, and after this, the patient data is received from the *Data Manager*, in *HL7 CDA* document format, it is parsed and displayed (Fig. 3.12).

In order to make the parsing, "*System.XML*" library from *.Net* is used.

As a response to the call made from "Search" page, the *Data Manager* sends back not only the *HL7 CDA* document containing the patient data, but the *HL7 CDA Level 2* document containing the medical recommendations made by the inference engine. This information (recommendations), after being parsed is displayed on other page. On this page, the medical staff can also see the patient data in parallel with the recommendations (Fig. 3.13).


Generarea automată a protocoalelor medicale

Recomandările medicale sunt:

Repetarea vaccinului antitetanus*

Repetarea vaccinului astazi*

Administrare Vaccin TD*

Administrarea acestuia impreuna cu tetanus*

Aplicarea unei injectii calmante*

Aplicarea acesteaia daca nu are contraindicatii*

Chestionarea pacientului despre nivelul glicemic*

Analizarea nivelului glicemic*

Datele pacientului

Medic curant: Dr. Ionescu Gheorghe

Sex: M

Data nașterii: 19600707

Probleme active: Obezitate, Diabet

Istoric medical: Psoriasis

Intervenții chirurgicale: Extirpare apendice

Alergii: Intoleranta la lapte, Alune, Penicilina

Antecedente familiale: Diabet

Vaccinuri: Tetanus în data: 19900514, Febra aftoasa în data: 20041104, Peneumon C-9 în data: 19660511

Obiceiuri sociale: Consum de tutun

Rezultate de la borator: Analiza de urina în data: 20051108, valoare: 6, unitate: %

Fig. 3.13. Patient data and recommendation

Another component which implements a well appreciated feature by the medical staff regards the possibility to enter a feedback (about the content of the recommendations or about the different features of the interface). This can help the further development, not only of the interface, but also of the knowledge base, by expressing opinions about the validity of the recommendations (results of the inference).

The communication between the service provider (*Data Manager*) and this web application (interface) is made with *XML* files. Regarding the server, it implements a Web service which responds to web application demands by sending *XML*s with requested data (patient data and recommendations), data that are extracted subsequently with specific functions.

In order to populate the web pages with data, the *XML*s are parsed then the specific information is acquired. The *XML* data are parsed by implementing the functionalities offered by *XPath* namespace in the *.Net* Framework. Each interface has associated C# functions in order to perform the data integration.

The implementation of such a solution resulted in an interface which integrates various features of the existing solution, in order to make medical staff and patients more compliant with the decision making support.

## 3.6. Data privacy and security

Regarding privacy/integrity issues, our system should achieve the *HIPAA* (Health Insurance Portability and Accountability Act) requirements; the first step would be implementing the communication over *HTTPS.* In the next section an analysis of *HIPAA* is made.

### ***HIPAA***

In order to make our architecture functional in the real word, some solutions are need for the security and privacy of the data. *HIPAA* can be used as a set of requirements for our system in order to accomplish data security and privacy.

The Health Insurance Portability and Accountability Act (*HIPAA*) is a USA law, which constrains the Department of Health and Human Services (*HHS*) to use standards for electronic exchange of medical information and identifiers for health plans, providers and employees [99].

It is very important for the patient that his medical information remains confidential and private. The rights, rules and limitations on the permission to see and receive health information about a patient are given by the *Privacy Rules*, which is a Federal law. Another part of *HIPAA* is represented by *Security Rules,* also a Federal law which is concerned with security of health information in electronic form.

Following is presented the way in which *Privacy Rules* are applied for General Public Health Activities and for other public health activities.

For General Public Health Activities it is permitted to access limited protected health information, without special authorization in order to prevent and control diseases, disabilities, or injuries. The amount of health information accessed by the public health authority (represented by an agency or authority of the US government, a State, a political subdivision of a State or a territory) is minimal, being limited to necessary information to protect the public health [100].

For other Public Health Activities it is permitted to access this health information if there are doubts: on child abuse or neglect, about quality, safety or effectiveness of a product or activity, about persons that present risk of contracting or spreading a disease or about workplace medical surveillance [100].

Several of the *HIPAA* advantages are [101, 102]:

- Assures privacy and protection of health records;
- Assures penalties if the privacy of health records is violated;
- Facilitates the providers work, because the health information will be processed in a common format in the US;
- Facilitates the health information for persons who change or lose their work;
- Limits the exclusions of pre-existing conditions;

- It forbids the discrimination between employees based on their health condition;
- The patients have the warranty of renewability and availability of health coverage.

As specified in [103, 104] there are several main disadvantages regarding *HIPAA* use and implementation:

- in [103] is shown the fact that although the *HIPAA* regulation improves patient data protection, there are still issues regarding the access to patient information for medical research (based on patient data) purposes, as well as regarding the time and founding needed for research;
- for the implementation of *HIPAA* in medical units it is needed that all the staff to be involved [104];
- there is also a problem in understanding what exactly *HIPAA* standards can represent for their medical unit [104].

Other problems related to *HIPAA* are highlighted in [105, 106]:

- the inefficiency in suing a company in case of *HIPAA* violation;
- lack of enforcement regarding the implementation of *HIPAA*;
- extra money that are needed for the specialized staff for the implementation and maintenance of *HIPAA;*
- multiple institutions (e.g., accountants, lawyers, billing services) can have access to patient data;
- *HIPAA* offers same protection for the current information as for information contained in old records, not offering extra protection;
- implementing *HIPAA* would increase costs.

Several solutions (implementing *HIPAA*) were suggested: key management for the protection of medical images in [107], the implementation of multiple cryptographic algorithms to ensure the safety of information (especially for the emergency cases) in [108], etc.

The motivation of this *HIPAA* analysis is that our solution offers small security for patient data. Based on this analysis, although *HIPAA* presents a series of disadvantages, implementing *HIPAA* for the presented system may increase its acceptance by offering a high degree of security for the patient information an increases.

## 3.7. System workflow

Based on *HL7 CDA Component* and *TM-vMR*, the decision-making system has access to clinical data from different institutions that can thus provide more accurate medical recommendations based on more complex information.

To obtain medical recommendations the system uses:

- the data received from *HL7 CDA Component* or from the *vMR* through the *Data Manager;*
- medical guidelines represented in *Arden Syntax* as a source of knowledge for the expert system [21].

In the next sections, the inputs of the systems as well as the inputs for each component of the system are presented. Based on this inputs the interaction of the various components is presented and how the information is transmitted from one component to the other. In this manner, one can see how the components deal (modify) with the data.

### 3.7.1. Inputs and outputs of the system components

Regarding the inputs of the system, they are:

- *MLMs* used as representation of medical rules;
- databases (and other patient data sources) used by *HL7 CDA Component* and *vMR*;
- patient ID

In table 3.1, all the inputs and outputs of each system component are presented. Each line represents a data type which is used as input or output by the components (each column representing a component)

| | *HL 7 CDA Component* | | *TM-vMR* | | *Data Manager* | | *Egadss* | | *Interface* | |
|---|---|---|---|---|---|---|---|---|---|---|
| | *In* | *Out* | *In* | *Out* | *In* | *Out* | *In* | *Out* | *In* | *Out* |
| *Patient data source ** | X | | X | | | | | | | |
| *HL 7 CDA I2* | | | | | X | X | | X | X | |
| *HL 7 CDA I3* | | X | | | X | X | X | | | |
| *vMR data* | | | | X | X | | | | | |
| *MLM* | | | | | | | X | | | |
| *Patient ID* | X | | X | | X | X | | | | X |
| *Medical recommendations* | | | | | | | | | | X |

Table 3.1. The inputs and outputs of the system components

* - any database or data inserted directly in the *TM-vMR*

In Fig. 3.14 is presented an exemplification of types of information used by each system component, in order to generate medical recommendations.

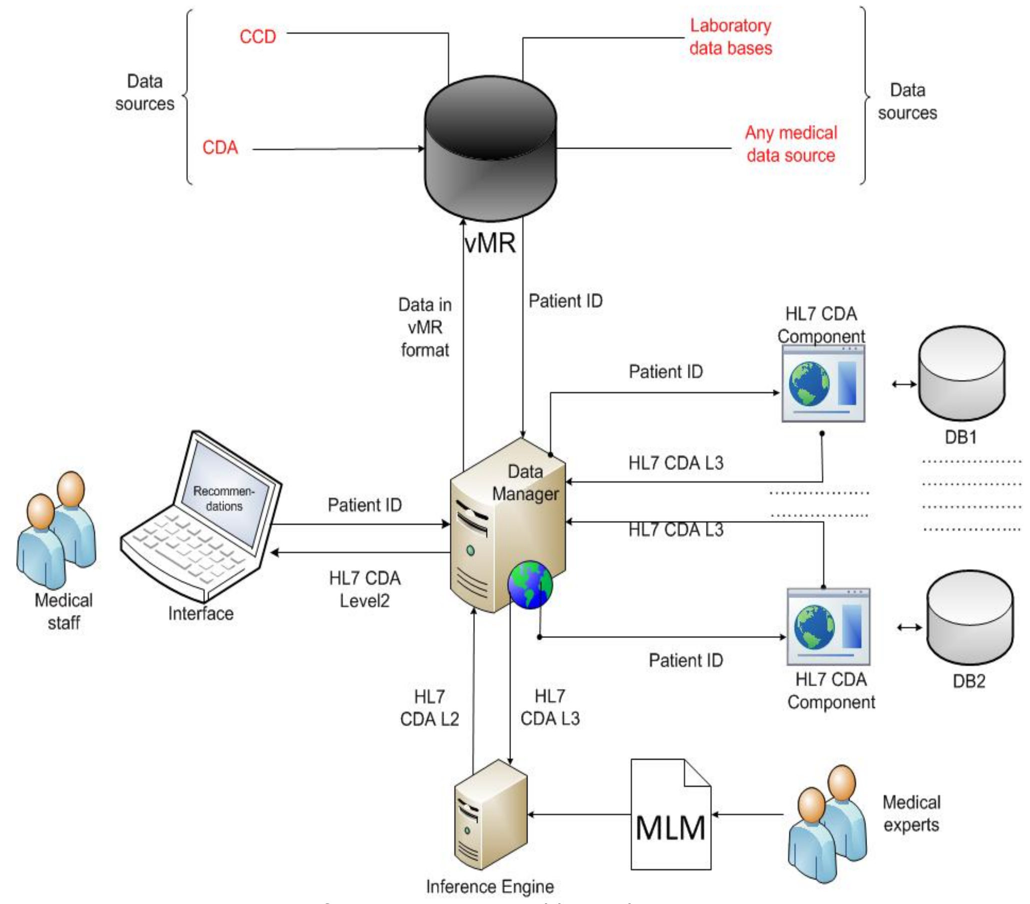


Fig. 3.14. Information types used by each system component

### 3.7.2. Data flows

Fig. 3.15 presents the sequence diagram for the developed system in order to obtain medical recommendations. It shows the interaction between objects in the sequential order when specific interactions occur.

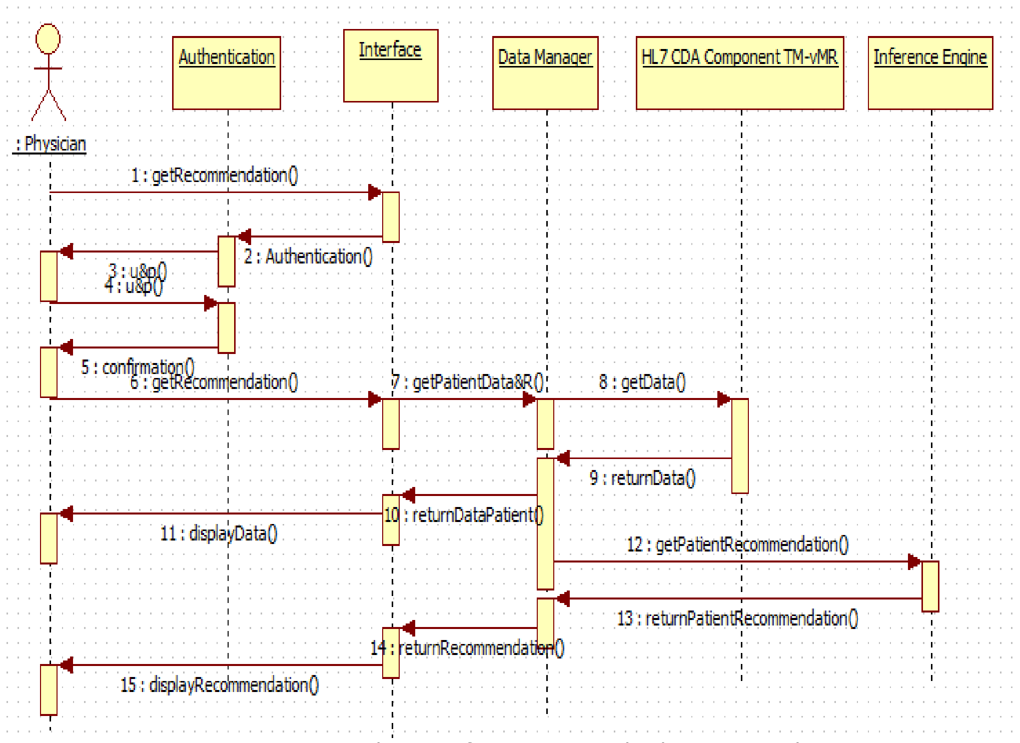


Fig. 3.15. Sequence diagram for getting medical recommendations

In Table 3.2 is presented the interaction which occurs in sequence diagram for generating medical recommendations.

| **Interactions** | **Description** |
|---|---|
| Interaction 1 | The physician wants to see the recommendations for a patient. |
| Interaction 2 | The doctor has to authenticate himself/herself in order to access data and recommendations. |
| Interaction 3 | Username and password are requested. |
| Interaction 4 | The physician sends the username and password. |
| Interaction 5 | The physician is logged into the system. |
| Interaction 6. | The physician wants to see the recommendations for a patient. |
| Interaction 7 | Request for the patient data and recommendation to the *Data Manager*. |
| Interaction 8 | Request for the patient data to the *HL7 Components* and *TM- vMR*. |
| Interaction 9 | *HL7 Components* and *TM- vMR* return patient data to the *Data Manager* |
| Interaction 10 | *Data Manager* returns patient data to the Interface |
| Interaction 11 | The physician visualizes the patient data. |
| Interaction 12 | Request for the recommendation to the inference engine. |
| Interaction 13 | Inference engine returns patient recommendation to the *Data Manager* |
| Interaction 14 | *Data Manager* returns patient recommendation to the Interface |
| Interaction 15 | The physician visualizes the recommendation for the patient. |

Table 3.2. System Workflow

## 3.8. Validation of the proposed solution

The validation of the software is made based on 30 patient dataset from the *Timişoara Emergency County Clinical Hospital*. Data types from the dataset are related to diabetes and they are:

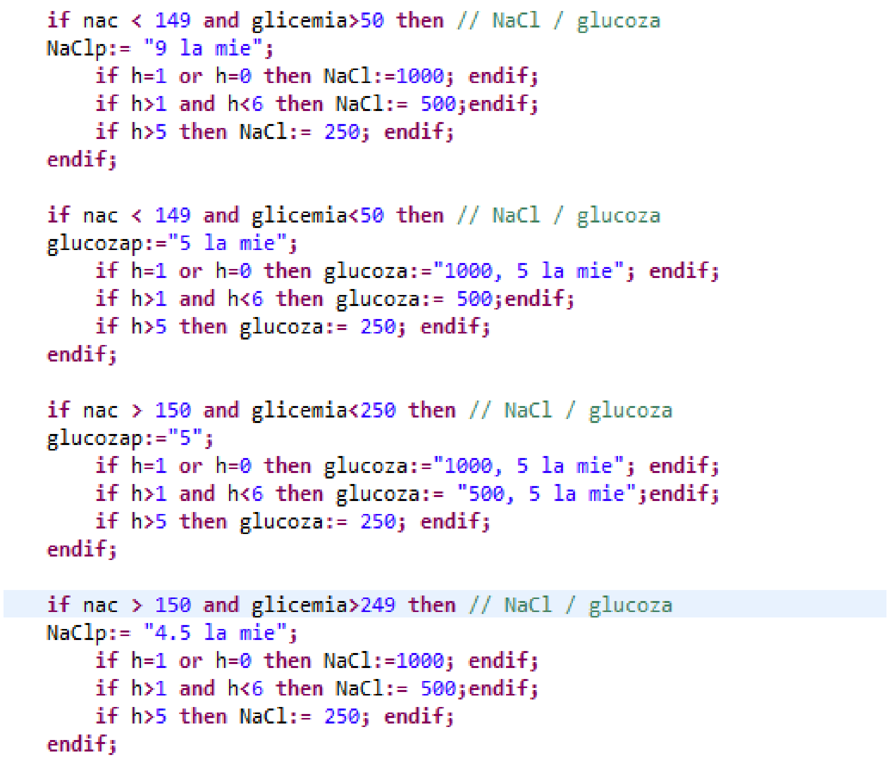
*Glycaemia* – is the amount of sugar present in the blood
*Alkaline Reserve* - the amount of reserve compounds (e.g. sodium bicarbonate);
*pH* – level pH in blood;
*Sodium* – level of sodium in blood;
*Potassium* – level of potassium in blood;
*weight* – weight of the patient;
*Urea* – amount of urea in blood;
*Ketones* – is true of false, for the presence of ketones.

The value intervals for these datasets are presented in Table 3.3.

| **Data Type** | **Values** | **Units** |
|---|---|---|
| *Glycaemia* | 40 – 350 | mg/dl |
| *Alkaline Reserve* | 6 – 26 | mmoli/L |
| *pH* | 6.8 – 7.45 | - |
| *Sodium* | 120 – 150 | mmol/L |
| *Potassium* | 2 – 7 | mmol/L |
| *weight* | 70 – 120 | Kg |
| *Urea* | 20 – 70 | mg/dl |
| *Ketones* | 1 / 0 | Boolean |

Table 3.3. Values of data regarding diabetes

The test was done for the diabetes management used in *Timişoara Emergency County Clinical Hospital*. *MLM*s where created, containing the medical rules. In Fig. 3.16 is represented a section of the Knowledge part of an *MLM* regarding the administration of *NaCl* or glucose.



```
if nac < 149 and glicemia>50 then // NaCl / glucoza
NaClp:= "9 la mie";
    if h=1 or h=0 then NaCl:=1000; endif;
    if h>1 and h<6 then NaCl:= 500;endif;
    if h>5 then NaCl:= 250; endif;
endif;

if nac < 149 and glicemia<50 then // NaCl / glucoza
glucozap:="5 la mie";
    if h=1 or h=0 then glucoza:="1000, 5 la mie"; endif;
    if h>1 and h<6 then glucoza:= 500;endif;
    if h>5 then glucoza:= 250; endif;
endif;

if nac > 150 and glicemia<250 then // NaCl / glucoza
glucozap:="5";
    if h=1 or h=0 then glucoza:="1000, 5 la mie"; endif;
    if h>1 and h<6 then glucoza:= "500, 5 la mie";endif;
    if h>5 then glucoza:= 250; endif;
endif;

if nac > 150 and glicemia>249 then // NaCl / glucoza
NaClp:= "4.5 la mie";
    if h=1 or h=0 then NaCl:=1000; endif;
    if h>1 and h<6 then NaCl:= 500;endif;
    if h>5 then NaCl:= 250; endif;
endif;
```

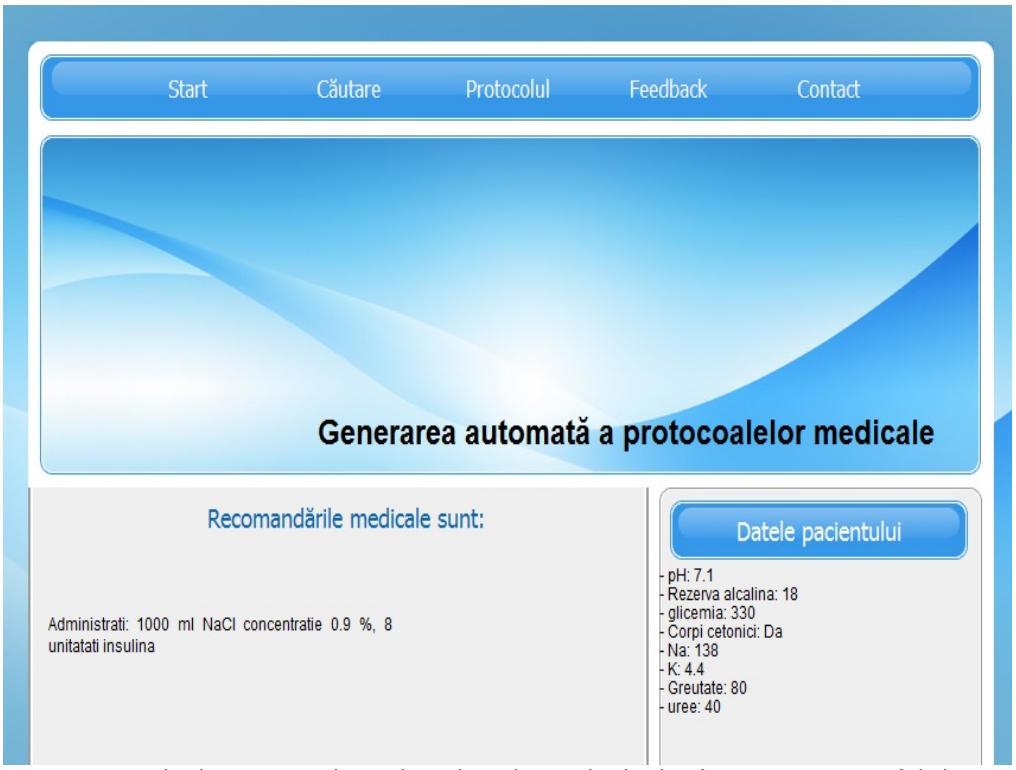
Fig. 3.16. Medical Rules in *Arden Syntax* for the prescription of *NaCl* or glucose

The steps of the testing method were:

- First, the 30 datasets were used for testing the system with the *CDA Component*.
- Based on this datasets, 60 *HL7 CDA Level 3* documents were created, 2 documents for each patient. This was done in order to simulate the existence of two different units, each unit having its own database.
- Then, from the interface, the ID of the patient was inserted.
- For each patient, based on the different *CDA* documents, the recommendation was generated.
- The recommendations previously generated, were evaluated by the physicians from the *Timişoara Emergency County Clinical Hospital*. They were considered correct for each dataset.
- In this way, the use of multiple *CDA Components* and *Data Manager* was validated.

Then the system was tested for the *TM-vMR* data source. Values from table 3.1 were inserted in the *TM-vMR* . This time, the PHP web services were called from the *Data Manager*. The method was similar.

In Fig. 3.17 is presented the recommendation which is made based on the *MLMs* regarding the management of the diabetes in the *Timişoara Emergency County Clinical Hospital*.



Recomandările medicale sunt:

Administrati: 1000 ml NaCl concentratie 0.9 %, 8 unitatati insulina

Datele pacientului

- pH: 7.1
- Rezerva alcalina: 18
- glicemia: 330
- Corpi cetonici: Da
- Na: 138
- K: 4.4
- Greutate: 80
- uree: 40

Fig. 3.17. Medical recommendation based on the medical rules for management of diabetes

## 3.9. Conclusions

I presented a solution which is based on *HL7* standards in order to improve the interoperability of *CDS* and clinical data sources. An architecture that has as inputs (for the different components) standardized data representation allows the integration of the entire system or only of some modules in different medical units.

The *Data Manager*, *HL7 CDA Component, TM-vMR*, *Egadss* and the *Interface* facilitates the communication between the components of the system and other medical information systems, allowing the access to multiple data source types.

Beside the use as inputs of the *HL7 CDA* documents, in order to increase the interoperability of the system, one of the *HL7* standards integrated in the system was *vMR*. The implementation of the *vMR* was done by developing a database using *Topic Maps* (taking the advantages of using cutting-edge technology) and developing the services needed to link this database to the *Data Manager* (*TM-vMR* link to *CDS*). The use of *Topic Maps* technology for the representation of medical data gives extensibility for the *vMR* data model. It keeps a low level of costs, as well, being developed with cheap (open-source) technologies and software. High levels of integration and compatibility with applications which implement *HL7* standards for interoperability of health information technology are also achieved. The software used to map the *vMR-DAM* (*Domain Analysis Model*) into a topic map is *Topincs*, which provides a very flexible and easy to learn/use environment.

Since target users are medical personnel, researchers from the healthcare and information technology domains, medical application developers, with little or no knowledge about *TM* standard, a special package of *Topincs* services was developed in order to avoid/eliminate the need of *TM* specialists. The development of the *topic map* followed the *vMR-DAM* as accurate as possible so the two models are virtually identical, and subsequent changes can be easily done.

The system was tested for protocols regarding diabetes management, for prescribing the dosage of: insulin, sodium chloride, potassium, glucose. The results of these tests met the physician's requirements.

Briefly, the contributions claimed by the author regarding the work presented in this Chapter are:

- Extending the *Egadss* capabilities by allowing the access to multiple *HL7 CDA* documents by adding a new module - *Data Manager* – meant to implement these special communication features.
- Development of an interface which implements some well appreciated features (specific for *CDSS*s) by physicians, based on a previous analysis of the solutions in the *CDS* domain.
- Implementation of the capability of the system to use *HL7 CDA* documents as data input.
- An analysis of the *HIPAA* regulation in other to implement a secure communication between the different components of the system with consequences in future enabling secure communication in my application.
- Development of methods and software solutions to connect the *CDSS* to the *TM-vMR* database.

# 4. USING *DATA MINING* RULES AND *ARDEN SYNTAX* IN THE PROCESS OF KNOWLEDGE DISCOVERY

The computer implementation of medical guidelines and protocols may not respond to all clinical situations, there might be cases when for a certain dataset, the *CDS* which implements clinical practice guidelines cannot offer an evidence based recommendation, but some medical advices can be generated based on knowledge discovery technologies. For this, we propose a solution which uses *data mining* technology for the discovery of medical rules and *Arden Syntax* as the formalism for the representation of medical rules. The use of *data mining* leads to the discovery of new patterns in medical data [81, 109, 110, 111]. The new obtained rules can be used in any *CDS* which implements the *Arden Syntax* standard, as the solution presented in the previous Chapter.

In this section, we present a solution for the extraction of medical rules from databases using *data mining* technology (with the open-source software *WEKA*) and the translation of these rules in *Arden Syntax* language.

In the next section is presented the architecture of the proposed system. In section 4.2 we can see the methods used for generating medical rules starting from a data base and using different *data mining* algorithms and *WEKA* open-source. Section 4.3 presents the method to translate *data mining* rules to *Arden Syntax* starting from parsing the rules until the generation of different slot of a *MLM* file. The development of the methods and their implementation (using *.Net* platform) is also presented. The results of using the developed system are presented in section 4.4. In section 4.5 some conclusions are drawn.

## 4.1. System components and architecture

In this section, is presented the architecture of a system used for the extraction of medical rules from data bases. This system (Figure 4.1) is based on two major components: the rule extraction part (based on *data mining*) and the translator to *Arden Syntax* language, which will be further referred as *DB2AS* (Database to *Arden Syntax*) [112]. The rule extraction part is based on *data mining* (for analyzing data and generating medical rules).

The translation part refers to the translation of "if-then" rules (in the format obtained by using *data mining*) to the *Arden Syntax* language. In order to obtain *Arden Syntax* rules from medical data the next steps are performed [113]:

- analyzing medical data,
- applying *data mining* algorithms,
- generation of medical rules,
- identification of the elements of the rules,
- *XML* representation of the rule elements (to be transmitted to the translation engine),
- use of rules elements for the generation of *Arden Syntax* rules.

Each step will be further presented in the next sections.

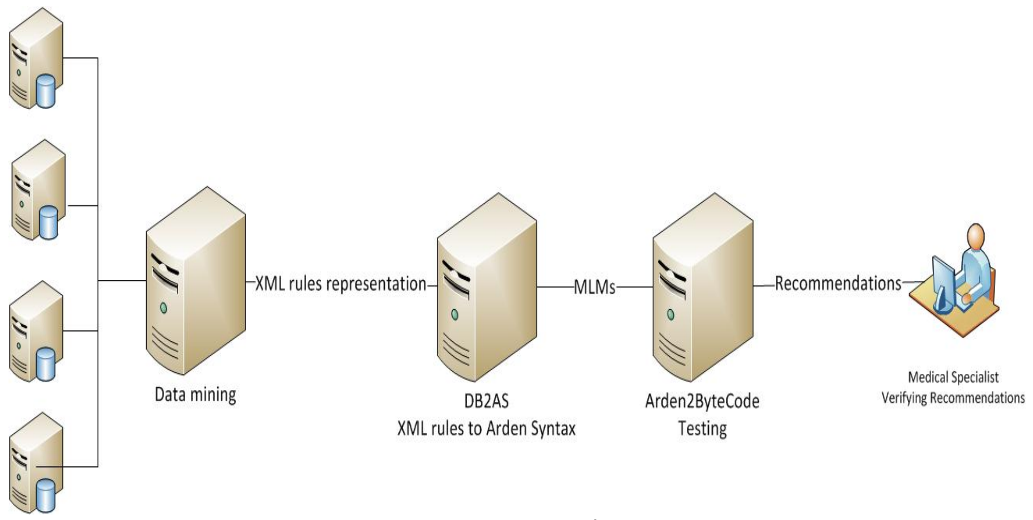


Figure 4.1. System architecture

## 4.2. From databases to *data mining* rules

The data source for discovering classification rules is represented by data regarding the 2326 births that took place in 2010 at the *Bega Obstetrics – Gynecology Hospital, Timişoara, Romania.* The algorithm that was used to discover the rules is *Ant Colony Optimization* from *Ant Miner* software [114].

Data regarding the mother, the fetus and the medical interventions performed on the mother or baby in order to help the birth were received as a Sheet in *Microsoft Excel* (there was no confidential data). For each birth the following data were retained:

- Age;
- The mother's background – rural or urban;
- Gesta – the number of the current pregnancy;
- Para – the number of the current birth ;
- The number of gestation weeks;
- The month in which the birth took place;
- The number of labor hours;
- Presentation - the fetus position at birth (with the possible values: cephalic – the fetus has its spine parallel the mother's and its head down, its chin in the chest, pelvian – the fetus comes out feet or bottom first, facial – the fetus looks ahead, its face comes out first and transversal – the fetus is positioned sidelong in the belly);
- Sex;
- Weight;
- Type of birth - natural birth or caesarean section;
- Videx – if a metal cap is used or not to help natural birth;

- Episiotomy – indicates if a cut was made in the perineum in order to help the natural birth;
- Emp – indicates if a manual extraction of the placenta was made;
- The Apgar score - which is an estimate of the vital functions and of the capacity to adapt to the conditions of the extra uterine environment, it is a score from 0 to 10.

The data were first pre-processed. The pre-processing consisted of:

- Eliminating data which was not interesting for this study (the month when the birth took place, the reason for the caesarean);
- Analyzing each attribute and correcting the data that were wrongfully introduced (for example for 2 babies the weight at birth was set at 34000 g, respectively 34450 g, we supposed that an extra 0 was introduced by mistake and it was eliminated) or eliminating them (in the *Apgar* score column, for two persons, instead of the score, we found the reason for which it was not given – birth at home or birth in the ambulance – the two instances were eliminated).

After eliminating the instances that contained missing or wrong data, we retained 2277 valid instances (from 2324) and 14 attributes for each pregnancy.

Then, the data were converted from the *xls* format into *arff* (*Atribute Relation File Format*) using *Arff Convertor* software tool [115]. Data in *arff* were loaded in *WEKA* [116] and the numeric ones such as the age of the mother, the weight of the baby, the number of labor hours, the number of gestation weeks were discredited by dividing these values into 5 equal intervals, saved and loaded in *Ant Miner* [114]. The discretization was necessary because the *Ant Colony Optimization* algorithm from *Ant Miner* works only with nominal data.

Initially, for the class attribute - *Apgar* score - 11 possible values were considered (values from 0 to 10). The performances of the classifier that was obtained in this manner were very weak. We returned to the pre-processing stage and the number of values for the *Apgar* score was reduced, transforming into a numeric attribute and then discrediting it with *WEKA* dividing it into 5 equal intervals. The same thing was done with the *gesta* and *para* attributes which had a large number of nominal values. The final form of the entry data for which a medium accuracy of the prediction of 78.74 % was obtained is presented next.

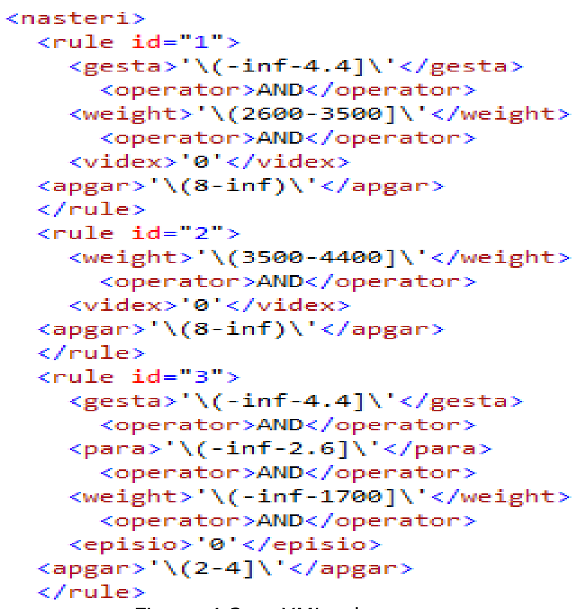
```
        @attribute age {'\'(-inf-18.6]\'','\'(18.6-25.2]\'','\'(25.2-31.8]\'','\'(31.8-
38.4]\'','\'(38.4-inf)\''}
        @attribute background {U,R}
        @attribute gesta {'\'(-inf-4.4]\'','\'(4.4-8.8]\'','\'(8.8-13.2]\'','\'(13.2-
17.6]\'','\'(17.6-inf)\''}
        @attribute para {'\'(-inf-2.6]\'','\'(2.6-5.2]\'','\'(5.2-7.8]\'','\'(7.8-
10.4]\'','\'(10.4-inf)\''}
        @attribute gestation_weeks {'\'(-inf-24.1]\'','\'(24.1-29.2]\'','\'(29.2-
34.3]\'','\'(34.3-39.4]\'','\'(39.4-inf)\''}
        @attribute presentation {CEF,PEL,TRANSV,FACIALA}
        @attribute sex {M,F}
        @attribute weight {'\'(-inf-1700]\'','\'(1700-2600]\'','\'(2600-
3500]\'','\'(3500-4400]\'','\'(4400-inf)\''}
        @attribute birth_type {N,C}
        @attribute videx {0,1}
        @attribute episio {0,1}
        @attribute hours_labour {'\'(-inf-6]\'','\'(6-12]\'','\'(12-18]\'','\'(18-
24]\'','\'(24-inf)\''}
        @attribute emp {0,1}
        @attribute Apgar {'\'(-inf-2]\'','\'(2-4]\'','\'(4-6]\'','\'(6-8]\'','\'(8-inf)\''}
```

The discovered rules were tested through cross validation. The initial dataset was divided in 10 subsets and the algorithm was run 10 times. Each time a different dataset was used for testing the accuracy of the discovered rules and the other nine subsets were used for training. The medium accuracy obtained by the rules is of 78.74 %. An example of the rules discovered which obtained on the test-set an accuracy of 85 % is presented:

```
        IF gesta = '\(-inf-4.4]\' AND weight = '\(2600-3500]\' AND videx = '0' THEN
'\(8-inf)\'
        IF weight = '\(3500-4400]\' AND videx = '0' THEN '\(8-inf)\'
        IF gesta = '\(-inf-4.4]\' AND para = '\(-inf-2.6]\' AND weight = '\(-inf-1700]\'
AND episio = '0' THEN '\(2-4]\'
        IF weight = '\(2600-3500]\' THEN '\(8-inf)\'
        IF gesta = '\(-inf-4.4]\' AND gestation_weeks= '\(29.2-34.3]\' AND
presentation = 'CEF' AND sex = 'M' THEN '\(-inf-2]\'
        IF background = 'U' AND gesta = '\(-inf-4.4]\' AND para = '\(-inf-2.6]\' AND
presentation = x'CEF' AND labour_hours= '\(-inf-6]\' THEN '\(8-inf)\'
        IF background = 'R' AND presentation = 'CEF' THEN '\(8-inf)\'
        IF para = '\(-inf-2.6]\' AND sex = 'M' THEN '\(8-inf)\'
        IF gestation_weeks = '\(29.2-34.3]\' AND weight = '\(-inf-1700]\' THEN '\(4-
6]\'
        IF gesta = '\(-inf-4.4]\' AND gestation_weeks = '\(39.4-inf)\' AND weight =
'\(1700-2600]\' THEN '\(8-inf)\'
        IF background = 'U' AND para = '\(-inf-2.6]\' AND gestation_weeks = '\(34.3-
39.4]\' AND episio = '0' THEN '\(8-inf)\'
        IF type_birth= 'N' THEN '\(6-8]\'
        IF age = '\(18.6-25.2]\' THEN '\(6-8]\'
        IF gesta = '\(4.4-8.8]\' THEN '\(4-6]\'
        Default rule: \(6-8]\
```

The order of application of rules when generating recommendations is important, because the list of the discovered rules is ordered. For each instance from the test-set, the list of rules is run over, in the order in which they were discovered, verifying if the antecedent of the current rule covers the instance. Once a rule whose antecedent covers the instance is found, the class predicted by this rule is compared with the instance's class and if they coincide, the number of instances correctly classified increases and if they don't, the number of instances incorrectly classified increases. For the case in which none of the discovered rules can classify a certain instance, an implicit rule has been added, which classifies the instance as belonging to the majority class. The accuracy is calculated reporting the number of correctly classified instances to the total number of instances from the test-set.

Next, a program in C was developed, which read a text file that contains the rules presented in the above mentioned form and transforms it into *XML* (Figure 4.2) format to serve as entry for the *DB2AS* system. In Fig. 3 below we can view the rules transformed into *XML* format:



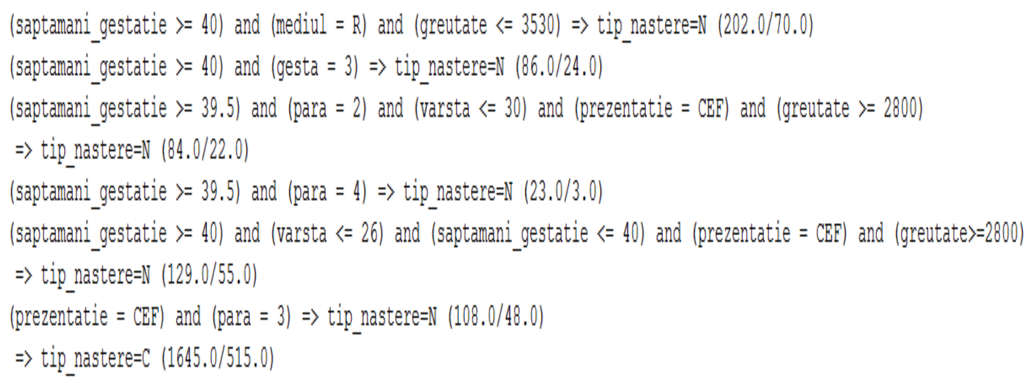
```
<nasteri>
  <rule id="1">
    <gesta>'\(-inf-4.4]\'</gesta>
      <operator>AND</operator>
    <weight>'\(2600-3500]\'</weight>
      <operator>AND</operator>
    <videx>'0'</videx>
  <apgar>'\(8-inf)\'</apgar>
  </rule>
  <rule id="2">
    <weight>'\(3500-4400]\'</weight>
      <operator>AND</operator>
    <videx>'0'</videx>
  <apgar>'\(8-inf)\'</apgar>
  </rule>
  <rule id="3">
    <gesta>'\(-inf-4.4]\'</gesta>
      <operator>AND</operator>
    <para>'\(-inf-2.6]\'</para>
      <operator>AND</operator>
    <weight>'\(-inf-1700]\'</weight>
      <operator>AND</operator>
    <episio>'0'</episio>
  <apgar>'\(2-4]\'</apgar>
  </rule>
```

Figure 4.2. – *XML* rules

For each new-born, it is desirable to have the best transition to the extra uterine environment, which takes shape into an *Apgar* score as high as possible. The built model can be used to make predictions with a trust of 78.9 % of the interval in which the *Apgar* value will be situated based on data regarding the mother, the baby and the medical interventions. Before the birth takes place, simulations can be made in order to see how high the *Apgar* score will be, information which could help the doctors decide the best measures in order to assure the easiest transition for the baby to the extra uterine environment.

Beside the rules obtained for the *Apgar* score*,* other rules were obtained based on other algorithms for the same database. This model can be used to make predictions with a trust less than 80 %. These new rules regard:

A. **Caesarean recommendation**; for this set, 6 rules (Figure 4.3) were discovered by using *JRIP* algorithm [117] in *WEKA*, obtaining medical rules with a low accuracy of 63.54%,
B. **The risks at birth** (no risk, medium risk, high risk): for this set, 16 rules were discovered, with accuracy of rules of 77%; *Ant Colony Optimization* algorithm from *Ant Miner* software was used.



```
(saptamani_gestatie >= 40) and (mediul = R) and (greutate <= 3530) => tip_nastere=N (202.0/70.0)
(saptamani_gestatie >= 40) and (gesta = 3) => tip_nastere=N (86.0/24.0)
(saptamani_gestatie >= 39.5) and (para = 2) and (varsta <= 30) and (prezentatie = CEF) and (greutate >= 2800)
 => tip_nastere=N (84.0/22.0)
(saptamani_gestatie >= 39.5) and (para = 4) => tip_nastere=N (23.0/3.0)
(saptamani_gestatie >= 40) and (varsta <= 26) and (saptamani_gestatie <= 40) and (prezentatie = CEF) and (greutate>=2800)
 => tip_nastere=N (129.0/55.0)
(prezentatie = CEF) and (para = 3) => tip_nastere=N (108.0/48.0)
 => tip_nastere=C (1645.0/515.0)
```

Figure 4.3. *Data mining* rules for caesarean recommendation

## 4.3. *DB2AS*

*DB2AS* module has as input the *XML* representation of rules received from *data mining* module. The outputs of the *DB2AS* module are *MLM* files. The components of the *DB2AS* for generating *MLM* are:

- *XML* parser,
- *Maintenance* and *Library* generator,
- *Knowledge* generator.

The implementation of *DB2AS* is done using *.Net* platform with *C#*.
The components are described in the following subsections:

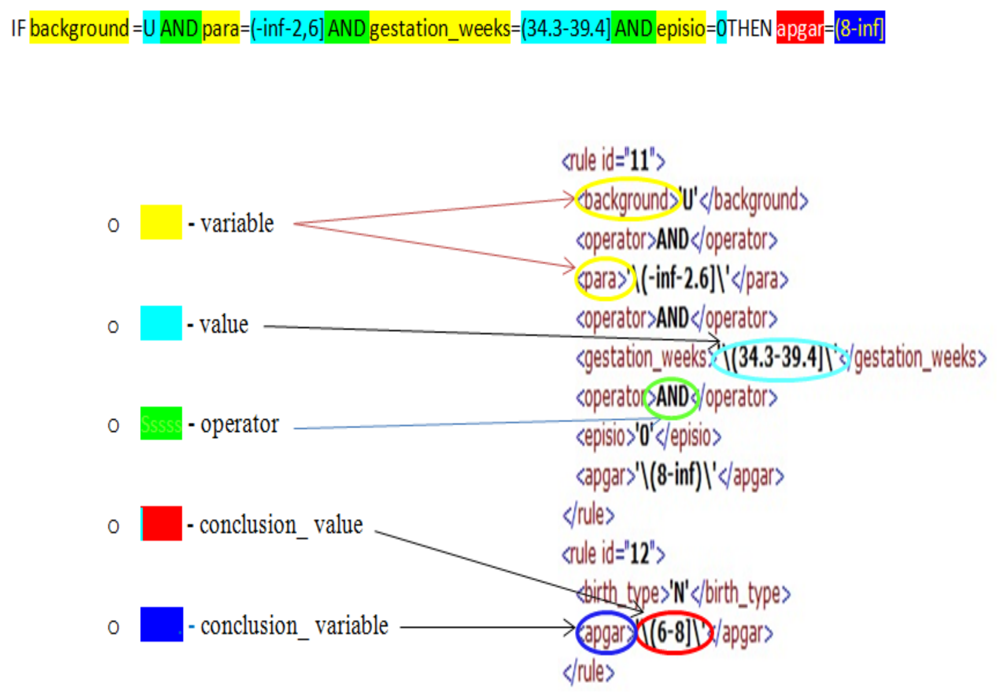
### 4.3.1. Data categorization and information extraction from the *XML*

When generating the different categories of the *MLM*, only some elements of the medical rules are needed for each category. Therefore, it was necessary to split the rules in several token types (e.g., *data* slot needs only the variable and conclusion_varible token types).

The first steps performed by the *DB2AS* are to extract and to categorize the information from the *XML*. The *XML* representation is used for the communication between the *data mining* rule extraction module and the *DB2AS.* An *XML* representation of the tokens can be visualized in Figure 4.

After the medical rules are obtained using *data mining* technology and based on a lexical analysis, the elements of the medical rules are categorized in five tokens types (Figure 4.4):

- variable;
- value;
- operator;
- conclusion_variable;
- conclusion_value.



```
IF background =U AND para=(-inf-2,6] AND gestation_weeks=(34.3-39.4] AND episio=0THEN apgar=(8-inf]
```

```
<rule id="11">
  <background>'U'</background>
  <operator>AND</operator>
  <para>'\(-inf-2.6]\'</para>
  <operator>AND</operator>
  <gestation_weeks>'\(34.3-39.4]\'</gestation_weeks>
  <operator>AND</operator>
  <episio>'0'</episio>
  <apgar>'\(8-inf)\'</apgar>
</rule>
<rule id="12">
  <birth_type>'N'</birth_type>
  <apgar>'\(6-8]\'</apgar>
</rule>
```

Figure 4.4. The elements of the medical rules categorized in tokens types

In Figure 4.4 we can see the correspondence between data elements of the *XML* and tokens. The first line represents the *data mining* rule. The elements of this rule are highlighted with different colors, each of those colors representing a token type. These token types can be found also in the *XML* representation of the rules (the right part of the figure). There was no need to tokenize the other components ("if", "then" or "=") of the "if-then" *data mining* rules.

The information from the *XML* is extracted in five different arrays representing the five token types (*variable, value, operator, conclusion_variable, conclusion_value*). The extraction of the different data elements is realized using the functions of the "System.*XML*" namespace from *.Net* platform. The elements of the arrays (tokens) are further processed in order to be integrated in the different *MLM* slots.

### 4.3.2. Use of the token types in *Arden Syntax*

In Table 4.1 is presented the *data mining* to *Arden Syntax* translation by the use of the five categories of data (*variable, value, operator, conclusion_variable, conclusion_value*) in the slots of an *MLM* is presented. There are also a few *MLM* slots that do not require data from the medical rules:

- for Maintenance *MLM*'s category: title, *MLM*name, slot, version, institution, author, specialist, date, validation;
- for Library *MLM*'s category: purpose, explanation;
- for Knowledge *MLM*'s category: type, evoke, action.

Some of these slots can be empty and some are completed with information such as:

- information generated based on information regarding the database, system, authors or applied *data mining* algorithms (e.g., value of *MLMname* is based on the name of the database used for the extraction of the medical rules) see section 4.3.3;
- default values;
- computed values in the *MLM* (i.e. the printed value from the *action* slot).

| | ***Keywords (Library)*** | ***Data (Knowledge)*** | ***Logic (Knowledge)*** |
|---|---|---|---|
| ***variable*** | YES | YES | YES |
| ***value*** | NO | NO | YES |
| ***operator*** | NO | NO | YES |
| ***conclusion_variable*** | YES | YES | YES |
| ***conclusion_value*** | NO | NO | YES |

Table 4.1. Presence of the five data categories from the medical rules used in the *MLM*

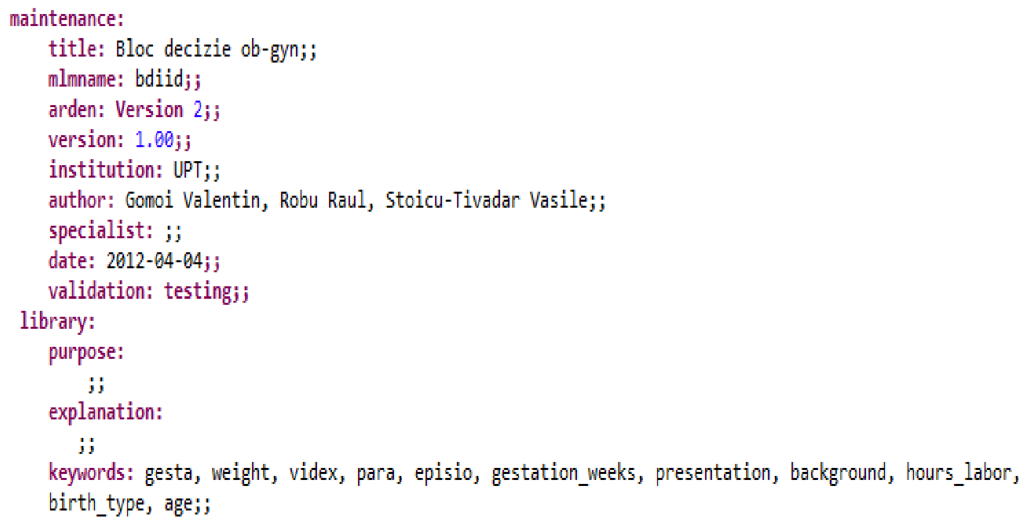
### 4.3.2. Maintenance and Library generator

The *Maintenance* and the *Library* categories contain the *MLM* documentation. The slots for *Maintenance* are generated based on the information related to the database (to which *data mining* algorithms are applied), system, authors or the applied *data mining* algorithms:

- *title* slot - based on the name of the database used for the extraction of the medical rules – e.g. Bdiid (the name of data base);
- *MLM*name slot - based on the name of the database used for the extraction of the medical rules – e.g. Bdiid;
- *version* slot - default value "1.0";
- *institution* slot - default "Politehnica University Timisoara";
- *author* slot – "Gomoi Valentin, Robu Raul, Vasile Stoicu-Tivadar" (system developers);
- *specialist* slot - is empty - the *MLM* has to be validated by a medical specialist, her/his name can be added after the *MLM* is validated;
- *date* slot - the system date, when the *MLM* was generated;
- *validation* slot – default value: "testing".

Concerning the *purpose*, *explanation* and *citations* slots from the *Library* category, they are empty because the information extracted using *data mining* and the information related to the databases is not complex enough to generate a narrative description for the generated *MLM*. The information for the *description* slot and the *purpose* slot can be added by the medical specialist that is also validating the *MLM*.

The *keywords* slot is generated based on the *variable* and *conclusion_variable* tokens type (e.g., keywords: gestation_weeks, weight, sex, background, hours_labor, birth_type, age). An example of the *Maintenance* and *Library* categories generated by *DB2AS* is presented in Figure 4.5.



```
maintenance:
    title: Bloc decizie ob-gyn;;
    mlmname: bdiid;;
    arden: Version 2;;
    version: 1.00;;
    institution: UPT;;
    author: Gomoi Valentin, Robu Raul, Stoicu-Tivadar Vasile;;
    specialist: ;;
    date: 2012-04-04;;
    validation: testing;;
 library:
    purpose:
        ;;
    explanation:
       ;;
    keywords: gesta, weight, videx, para, episio, gestation_weeks, presentation, background, hours_labor,
    birth_type, age;;
```

Figure 4.5. *Maintenance* and *Library* categories generation

### 4.3.4. Knowledge category.

The *Knowledge* category contains the code necessary for the extraction of medical data, the medical rules and the action to be done (see section 2.2). Two of the five mandatory slots are generated without the use of the tokens:

- *type* – is implicit *data-driven*;
- *evoke* – has no arguments.

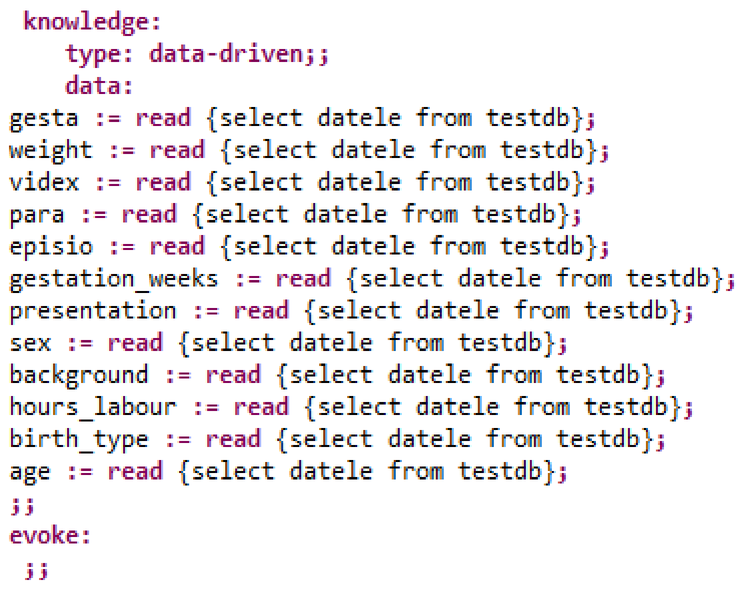
The action slot is based on a variable to which the result of each "if-then" rule is assigned. It contains a simple action, "write", that enforces the display of the solution.

The other two, more laborious to obtain slots are: *data*, and *logic*. The procedure for the generation of these slots is described below.

#### *A. Generating data slot*

The *data* slot contains the variables which are used in the "if-then" rules of the *logic* slot. The variables generation is based on the variables used in the "if-then" rules (Fig. 4) extracted with *data mining*.

In order to define all the variables needed in the *logic* slot, the *XML* containing all medical rules is parsed and every time a new variable is identified as part of any medical rule it is added to the: *variable* array or *conclusion_variable* (if it is the last variable in the rule) array. After all the variables are extracted, they are added to the *data* slot. Every variable has associated the next string: ":= read {};". The address for the extraction of the variable value must be specified by the *MLM* users in the curly brackets (Figure 4.6). This address is specific for a local context. In our implementation we added a default address for all the variables (e.g., „weight := read {select data from testdb};").



```
 knowledge:
    type: data-driven;;
    data:
gesta := read {select datele from testdb};
weight := read {select datele from testdb};
videx := read {select datele from testdb};
para := read {select datele from testdb};
episio := read {select datele from testdb};
gestation_weeks := read {select datele from testdb};
presentation := read {select datele from testdb};
sex := read {select datele from testdb};
background := read {select datele from testdb};
hours_labour := read {select datele from testdb};
birth_type := read {select datele from testdb};
age := read {select datele from testdb};
;;
evoke:
 ;;
```

Figure 4.6. *MLM* – *data* slot, *Knowledge* category

Testing the *MLM* with "*Arden2ByteCode*" [48] open-source software and using the address specified previously, the information could be extracted from "*testdb*" database or if the database was not present the data was inserted manually.

### *B. Generating logic slot*

The logic slot is the core of the *MLM* and it contains the medical rules. Based on these rules, a certain action is triggered. To generate medical rules all five arrays which were extracted from the *XML* (section 4) are needed (*variable, value, operator, conclusion_variable, conclusion_value*). This slot is entirely generated by the *DB2AS* module.

Building a rule from the different tokens has the next form:

if [variable]=[value] [operator] [variable]=[value] [operator]… [variable]=[value] then solution := "[conclusion_variable]=[ conclusion_value]"

The obtained result is:

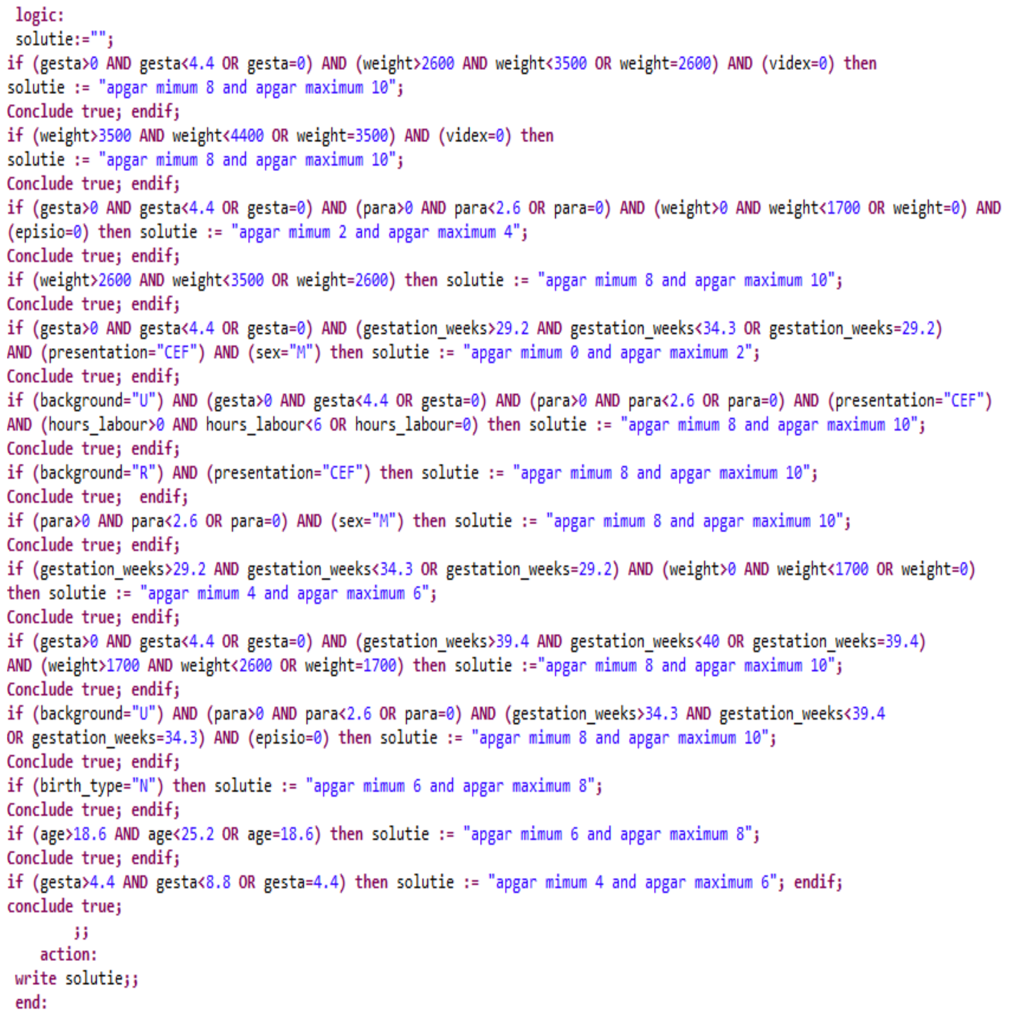
"**if (**gesta<4.4**) AND (**weight>2600 **AND** weight<3500 **OR** weight=2600**) AND (**videx=0**) then**
solution **:=** "*Apgar* minimum 8 AND *Apgar* maximum 10"**;**"

When composing a rule, the first steps are: adding the *variable* extracted from the *XML* (for each rule) then associating a *value* with this variable ("*variable-value*" association). The data type of the values is determined: number, date, string or interval.

To determine if the value is an interval, a special function is created. This function extracts the two values of the interval and the brackets. The variable has to be bigger than the first number of interval, smaller than the last number of interval or equals to a number of the interval if it is preceded by "[" or followed by "]" (e.g., "age=(31.8-38.4]" is translated in "age**>**31.8 **AND** age**<**38.4 **OR** age**=**31.8"). After a "variable – value" association is created, an *operator* ("and", "or") is added.

After the premise of the rule is generated, the *conclusion* of the rule is built. The *conclusion* of the rule is a string variable (the finale solution) to which is assigned a string containing the *conclusion_variable* and the *conclusion_value* (solution:="[conclusion_variable]=[ conclusion_value]"). The *solution* variable is used in the *action* slot.

These steps are repeated until all the rules from the *XML* are translated into *Arden Syntax* Language (Figure 4.7).



```
 logic:
 solutie:="";
if (gesta>0 AND gesta<4.4 OR gesta=0) AND (weight>2600 AND weight<3500 OR weight=2600) AND (videx=0) then
solutie := "apgar mimum 8 and apgar maximum 10";
Conclude true; endif;
if (weight>3500 AND weight<4400 OR weight=3500) AND (videx=0) then
solutie := "apgar mimum 8 and apgar maximum 10";
Conclude true; endif;
if (gesta>0 AND gesta<4.4 OR gesta=0) AND (para>0 AND para<2.6 OR para=0) AND (weight>0 AND weight<1700 OR weight=0) AND
(episio=0) then solutie := "apgar mimum 2 and apgar maximum 4";
Conclude true; endif;
if (weight>2600 AND weight<3500 OR weight=2600) then solutie := "apgar mimum 8 and apgar maximum 10";
Conclude true; endif;
if (gesta>0 AND gesta<4.4 OR gesta=0) AND (gestation_weeks>29.2 AND gestation_weeks<34.3 OR gestation_weeks=29.2)
AND (presentation="CEF") AND (sex="M") then solutie := "apgar mimum 0 and apgar maximum 2";
Conclude true; endif;
if (background="U") AND (gesta>0 AND gesta<4.4 OR gesta=0) AND (para>0 AND para<2.6 OR para=0) AND (presentation="CEF")
AND (hours_labour>0 AND hours_labour<6 OR hours_labour=0) then solutie := "apgar mimum 8 and apgar maximum 10";
Conclude true; endif;
if (background="R") AND (presentation="CEF") then solutie := "apgar mimum 8 and apgar maximum 10";
Conclude true;  endif;
if (para>0 AND para<2.6 OR para=0) AND (sex="M") then solutie := "apgar mimum 8 and apgar maximum 10";
Conclude true; endif;
if (gestation_weeks>29.2 AND gestation_weeks<34.3 OR gestation_weeks=29.2) AND (weight>0 AND weight<1700 OR weight=0)
then solutie := "apgar mimum 4 and apgar maximum 6";
Conclude true; endif;
if (gesta>0 AND gesta<4.4 OR gesta=0) AND (gestation_weeks>39.4 AND gestation_weeks<40 OR gestation_weeks=39.4)
AND (weight>1700 AND weight<2600 OR weight=1700) then solutie :="apgar mimum 8 and apgar maximum 10";
Conclude true; endif;
if (background="U") AND (para>0 AND para<2.6 OR para=0) AND (gestation_weeks>34.3 AND gestation_weeks<39.4
OR gestation_weeks=34.3) AND (episio=0) then solutie := "apgar mimum 8 and apgar maximum 10";
Conclude true; endif;
if (birth_type="N") then solutie := "apgar mimum 6 and apgar maximum 8";
Conclude true; endif;
if (age>18.6 AND age<25.2 OR age=18.6) then solutie := "apgar mimum 6 and apgar maximum 8";
Conclude true; endif;
if (gesta>4.4 AND gesta<8.8 OR gesta=4.4) then solutie := "apgar mimum 4 and apgar maximum 6"; endif;
conclude true;
        ;;
    action:
 write solutie;;
 end:
```

Figure 4.7. *MLM* – Knowledge category

## 4.4. Results

The system was tested to:

- see if the semantic of the *data mining* rules could be found in the *MLMs*;
- see if the new *MLM* can be integrated with the existing solution regarding the execution of *Arden Syntax* files;
- see if the advices are correct (based on the first *data mining* rules).

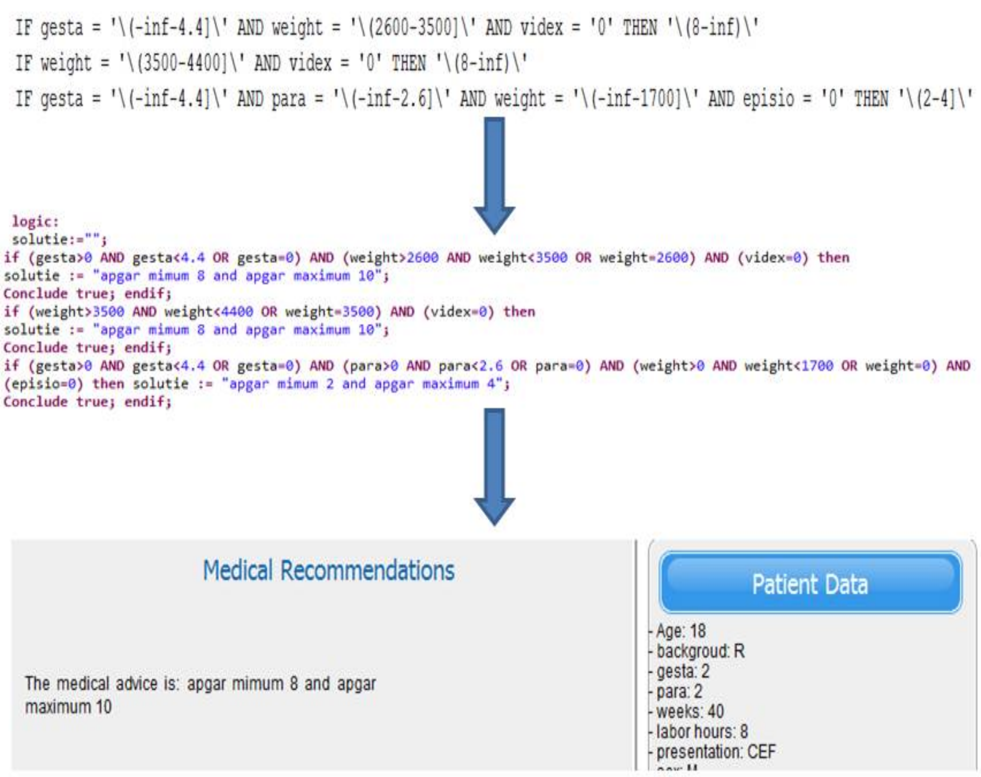
The system was tested for the newborns database mentioned in section 4.3. Three distinct types of rules which can be used to make predictions were obtained. Each of these types of rules was tested for the previous criteria.

### 4.4.1. Rules for *Apgar* score

Fourteen rules were obtained (see section 4.3). The resulted *Knowledge* category, for *Apgar* score, could be visualized in Figure 4.7, the resulted *Maintenance* and *Library* categories could be visualized in Figure 4.6.

After the *MLM* was generated (containing the 14 rules regarding the *Apgar* score), it was tested with *Arden2ByteCode* open-source software and with the system presented in Chapter 3. Different medical scenarios were elaborated based on the data from the newborns database. The new rules were validated for these scenarios (using 50 datasets containing information about newborns), the execution of the *MLM* offering the expected advices.

In Figure 4.8, we can see the steps from *data mining* rules to medical advices. The advice was made based on the generated *MLM* with the system presented in Chapter 3.



```
IF gesta = '\(-inf-4.4]\' AND weight = '\(2600-3500]\' AND videx = '0' THEN '\(8-inf)\'
IF weight = '\(3500-4400]\' AND videx = '0' THEN '\(8-inf)\'
IF gesta = '\(-inf-4.4]\' AND para = '\(-inf-2.6]\' AND weight = '\(-inf-1700]\' AND episio = '0' THEN '\(2-4]\'
```

```
 logic:
 solutie:="";
if (gesta>0 AND gesta<4.4 OR gesta=0) AND (weight>2600 AND weight<3500 OR weight=2600) AND (videx=0) then
solutie := "apgar mimum 8 and apgar maximum 10";
Conclude true; endif;
if (weight>3500 AND weight<4400 OR weight=3500) AND (videx=0) then
solutie := "apgar mimum 8 and apgar maximum 10";
Conclude true; endif;
if (gesta>0 AND gesta<4.4 OR gesta=0) AND (para>0 AND para<2.6 OR para=0) AND (weight>0 AND weight<1700 OR weight=0) AND
(episio=0) then solutie := "apgar mimum 2 and apgar maximum 4";
Conclude true; endif;
```

The medical advice is: apgar mimum 8 and apgar maximum 10

- Age: 18
- backgroud: R
- gesta: 2
- para: 2
- weeks: 40
- labor hours: 8
- presentation: CEF

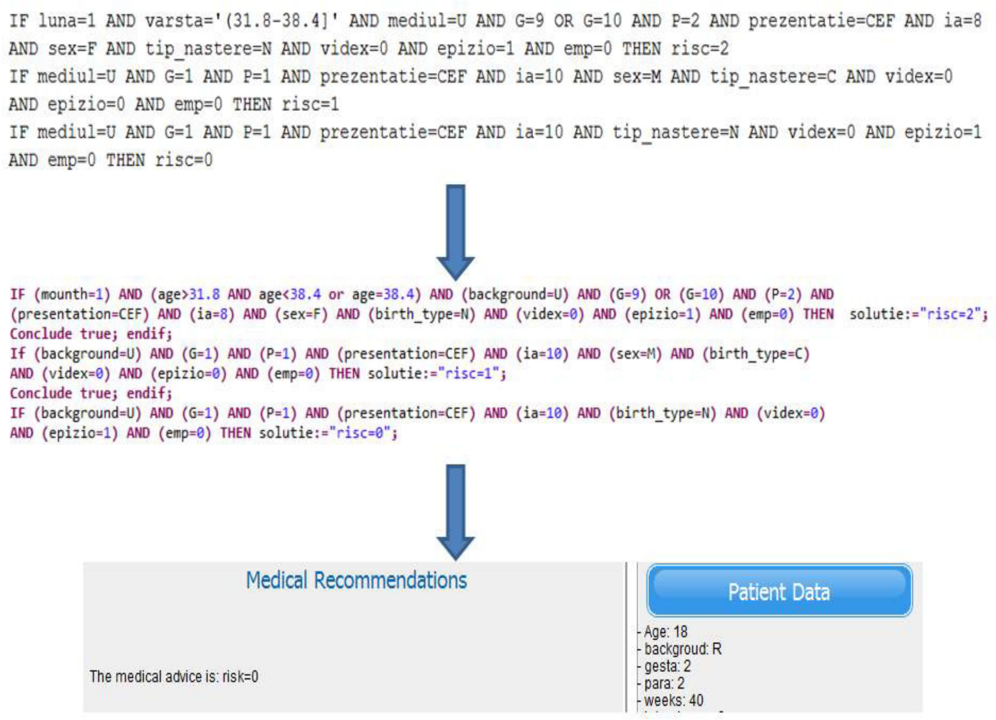
Figure 4.8. From d*ata mining* rules to medical advices for *Apgar* score

In the lower side of Figure 4.8, we can see the advice concerning the *Apgar* score that should be given to a newborn. Although the *Apgar* score is easily obtained using conventional methods, the generation of the medical advice for the *Apgar* score is used to verify the functionality of our system. Secondly, the generated medical advice can be used in order to have a confirmation that the correct *Apgar* score was given by the medical staff (maybe when calculating the *Apgar* score some aspects where not considered properly).

### 4.4.2. Rules for risk at birth

The *DB2AS* was further tested for the rules obtained for the risk at birth. After the *MLM* was generated (containing the 16 rules regarding the risk at birth), it was tested with *Arden2ByteCode* open-source software and with the help of the system presented in Chapter 3. Different medical scenarios were elaborated based on the data from the newborns database. The new rules were tested for these scenarios (using 30 datasets containing information about newborns), offering the expected advices.

In Figure 4.9 we can see the steps from *data mining* rules to medical advices. The advice predicting the risk at birth was made based on the generated *MLM* with the system presented in Chapter 3.



```
IF luna=1 AND varsta='(31.8-38.4]' AND mediul=U AND G=9 OR G=10 AND P=2 AND prezentatie=CEF AND ia=8
AND sex=F AND tip_nastere=N AND videx=0 AND epizio=1 AND emp=0 THEN risc=2
IF mediul=U AND G=1 AND P=1 AND prezentatie=CEF AND ia=10 AND sex=M AND tip_nastere=C AND videx=0
AND epizio=0 AND emp=0 THEN risc=1
IF mediul=U AND G=1 AND P=1 AND prezentatie=CEF AND ia=10 AND tip_nastere=N AND videx=0 AND epizio=1
AND emp=0 THEN risc=0
```

```
IF (mounth=1) AND (age>31.8 AND age<38.4 or age=38.4) AND (background=U) AND (G=9) OR (G=10) AND (P=2) AND
(presentation=CEF) AND (ia=8) AND (sex=F) AND (birth_type=N) AND (videx=0) AND (epizio=1) AND (emp=0) THEN  solutie:="risc=2";
Conclude true; endif;
If (background=U) AND (G=1) AND (P=1) AND (presentation=CEF) AND (ia=10) AND (sex=M) AND (birth_type=C)
AND (videx=0) AND (epizio=0) AND (emp=0) THEN solutie:="risc=1";
Conclude true; endif;
IF (background=U) AND (G=1) AND (P=1) AND (presentation=CEF) AND (ia=10) AND (birth_type=N) AND (videx=0)
AND (epizio=1) AND (emp=0) THEN solutie:="risc=0";
```

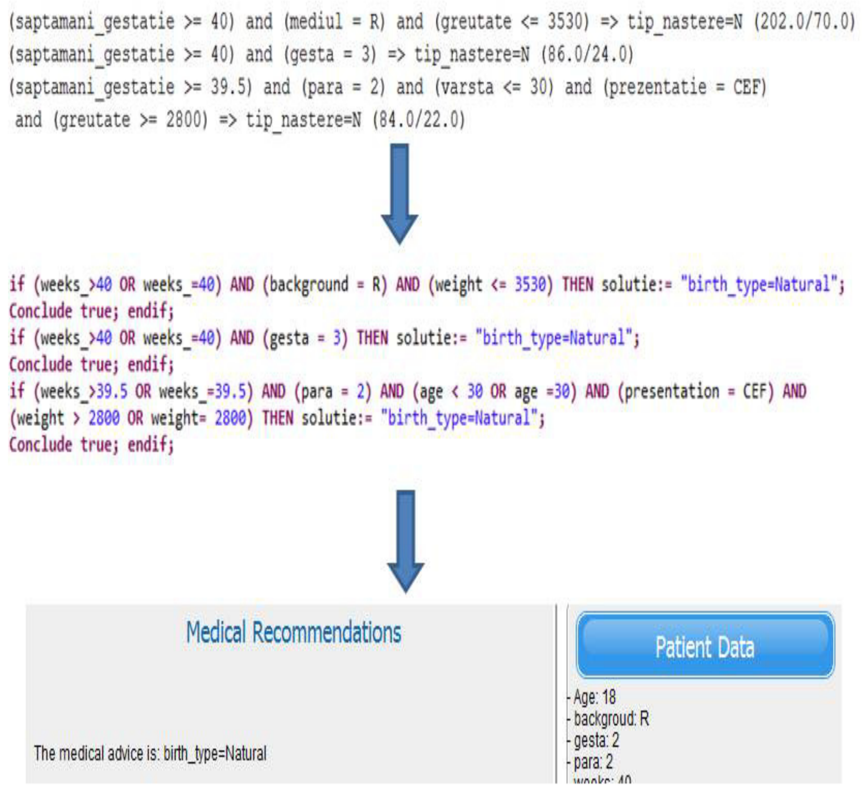
The medical advice is: risk=0

- Age: 18
- backgroud: R
- gesta: 2
- para: 2
- weeks: 40

Figure 4.9. From *data mining* rules to medical advices for risk at birth

### 4.4.3. Rules for predicting necessity of Caesarean section

The *DB2AS* was further tested for the rules obtained for the cases in which a caesarean section should be performed. After the *MLM* was generated (containing the 6 rules regarding the risk at birth), it was tested with *Arden2ByteCode* open-source software and with the help of the system presented in Chapter 3. Also the new rules were tested for these scenarios (using 30 datasets containing information about newborns), offering the expected advices (Figure 4.10). The advice regarding the necessity of performing a Caesarean section was made based on the generated *MLM* with the system presented in Chapter 3.



```
(saptamani_gestatie >= 40) and (mediul = R) and (greutate <= 3530) => tip_nastere=N (202.0/70.0)
(saptamani_gestatie >= 40) and (gesta = 3) => tip_nastere=N (86.0/24.0)
(saptamani_gestatie >= 39.5) and (para = 2) and (varsta <= 30) and (prezentatie = CEF)
 and (greutate >= 2800) => tip_nastere=N (84.0/22.0)
```

```
if (weeks_>40 OR weeks_=40) AND (background = R) AND (weight <= 3530) THEN solutie:= "birth_type=Natural";
Conclude true; endif;
if (weeks_>40 OR weeks_=40) AND (gesta = 3) THEN solutie:= "birth_type=Natural";
Conclude true; endif;
if (weeks_>39.5 OR weeks_=39.5) AND (para = 2) AND (age < 30 OR age =30) AND (presentation = CEF) AND
(weight > 2800 OR weight= 2800) THEN solutie:= "birth_type=Natural";
Conclude true; endif;
```

Medical Recommendations

The medical advice is: birth_type=Natural

Patient Data

- Age: 18
- backgroud: R
- gesta: 2
- para: 2

Figure 4.10. From *data mining* rules to medical advices regarding the necessity of Caesarean section

## 4.5. Conclusions

In this Chapter, I proposed, developed and implemented a new method and a set of associated applications for medical rules discovery and the formalization of these rules in *Arden Syntax,* in order to allow their use in various *CDSS*s. The *DB2AS* system was tested and verified for data regarding 2326 births that took place in 2010 at the *Bega Obstetrics – Gynecology Hospital, Timişoara, Romania*. Three different types of rules were obtained for predicting the *Apgar* score, the risk at birth and the necessity of Caesarean section.

The system is able to generate medical rules in standardized (*Arden Syntax*) manner, starting from datasets, with *data mining* technology. The conversion of *data mining* rules into *MLM,* enable the use of the new medical rules in multiple medical units implementing *Arden Syntax* standard. The adaptation of the resulted *MLMs* to a local context can be achieved by modifying the data source addresses in the curly brackets from the *data* slot. The new medical rules can help in situations in which the medical guidelines cannot offer solutions. The new medical advices can also be used as an alternative to the existing recommendation.

The most delicate issue regarding our system is represented by the acceptance of new medical rules by the medical staff. In this respect, each new *MLM* should be validated by medical experts. Testing the new medical advices in medical units represents the next step in improving their acceptance.

The accuracy of the prediction based on *data mining* rules is close to 80%. The issue regarding the accuracy will be addressed by future development of the system.

Briefly, the contributions claimed by the author regarding the work presented in this Chapter are:

- Categorization of the elements of the rules in: *variable, value, operator, csonclusion_variable, conclusion_value*. This categorization of the variables was necessary in order to generate the different *MLM's* sections.
- Development of a method which allows the translation of *data mining* rules into specific *Arden Syntax* files (*MLMs*) in order to make the knowledge discovered with *data mining* technology more accessible to *CDSS*s (through the representation of knowledge with standards).
- Development of a system architecture which implements all the previous methods.
- Development of all architecture modules needed (in C#).
- Analysis of the newborn database from *Bega Obstetrics – Gynecology Hospital, Timişoara, Romania* in order to select which data types are needed for the knowledge discovery;

Generation of medical rules (by using the developed system) in *Arden Syntax* format for the prediction of *Apgar* score, risks at birth and necessity of Caesarean section, validated by executing the resulted *MLMs* with the system presented in Chapter 3.

# 5. ACQUIRING MEDICAL KNOWLEDGE IN *ARDEN SYNTAX* FORMAT

In order to increase the interoperability of *CDSS*s, besides using standards for the representation of patient data (*Hl7 CDA*, [118, 119, 120]) we can use standards for the representation of medical rules (*Arden Syntax*).

In this chapter I compare the expressiveness of *ASTI 3* recommendation representation and *Arden Syntax* [27] representation by trying to identify the possibility of implementing the different components of *ASTI 3* recommendations and treatment models in *Arden Syntax* representation. I chose the *Arden Syntax* standard version 2.8 [27]. This version is the last one, accepted as a standard and is the most complex regarding the possibility of representing medical rules.

This comparison has as main objective the identification of a way to map *ASTI* medical rules in *Arden Syntax* in order to allow the exchange of medical knowledge between institutions. This mapping can have several of benefits:

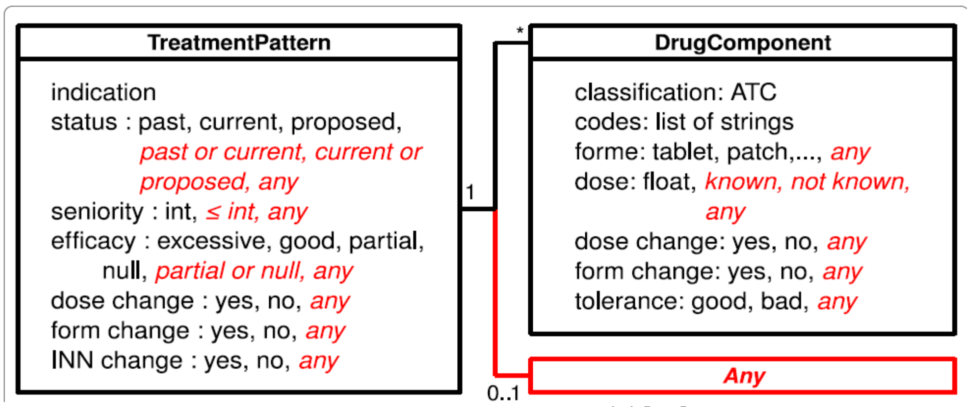
- increase the *Arden Syntax* knowledge base by allowing the translation of medical rules from other formalism
- increase the *ASTI* knowledge base by allowing the translation of medical rules from *Arden Syntax* formalism
- decrease the cost of *CPGs* computer formalization by direct translation from other formalism (fewer experts are needed)

This work was done at the University of Paris13, "Laboratoire d'informatique médicale et bioinformatique", Paris, France, department director: Alain Venot and in collaboration with Brigitte Seroussi and Harry Karadimas.

## 5.1. Challenges in the representation of *ASTI 3* recommendation in *Arden Syntax*

The first major challenge encountered in this comparison, between *ASTI 3* recommendations and *Arden Syntax* recommendations, is to identify the possibility of representing the drugs and the treatment from the *Treatment* model of *ASTI 3* in *Arden Syntax*. The *ASTI 3* representation of treatments is a complex, object oriented, based on the use of drug objects (*DrugComponent* in the model). Figure 5.1 shows the representation of the *ASTI* treatment model [73].

In this model we can see that a treatment can have multiple drugs associated. Each drug is defined by: *ATC (Anatomical Therapeutic Chemical) classification, codes, form, dose, dose change, form change, tolerance*. Every drug has at least one cod and cod type associated. The drug can also have other classification like SNOMED CT [121].


TreatmentPattern
indication
status : past, current, proposed, past or current, current or proposed, any
seniority : int, ≤ int, any
efficacy : excessive, good, partial, null, partial or null, any
dose change : yes, no, any
form change : yes, no, any
INN change : yes, no, any
DrugComponent
classification: ATC
codes: list of strings
forme: tablet, patch,..., any
dose: float, known, not known, any
dose change: yes, no, any
form change: yes, no, any
tolerance: good, bad, any
Any
*
1
0..1
Figure 5.1. *ASTI* treatment model [73]

Each treatment is also characterized by seven other attributes: *indication, status, seniority, efficacy, dose change, form change, INN (International Nonproprietary Name) change.*

The second major challenge in the comparison between *ASTI* recommendations and *Arden Syntax* recommendations, is defining the possibility of representing the medical rules of *ASTI 3* (Figure 5.2) in *Arden Syntax*.

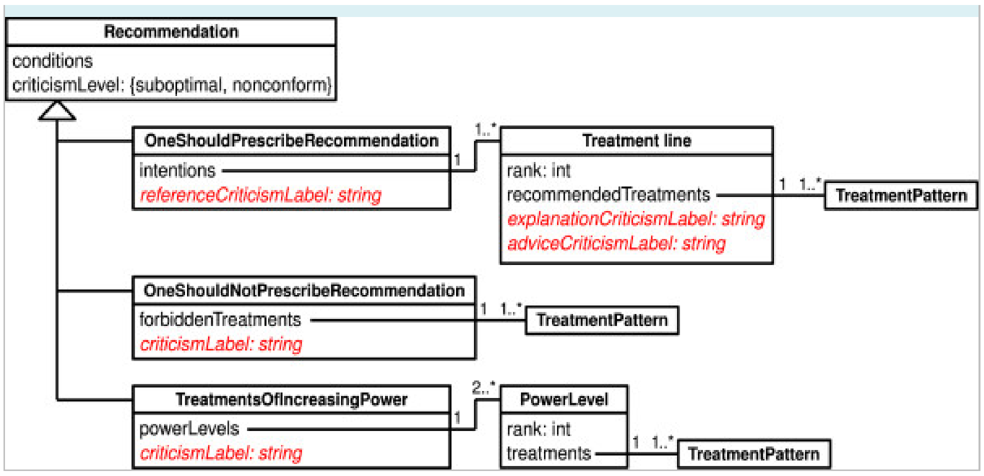


Figure 5.2. Recommendation model in *ASTI 3* [73]

For the comparison, a short description of the way in which the *ASTI 3* recommendations and treatment models are represented is needed. The representation implemented with *Python* language.

## 5.2. Implementing *ASTI 3* model with *Python*

The main steps in the representation of medical rules with the help of *ASTI 3* solution are presented in the following paragraphs:

***Variables and associated functions representation***

A set of needed variables (associated with the patient's status) must be declared (e.g., age, sex, HbA1c_max, HbA1c, weight). These variables are needed in order to make the inference.

Not all values associated to these variables are obtained directly from databases. Some functions are defined in order to compute values like *BMI* (body mass index) or to extract the maximum value from a list, values which are later used as inputs in the rules.

Medication representation

Following the medication is represented. An example of a medication representation is presented in the followings:

```
        glinide                = Composante("glinide"           , u"Glinide", ["A10BX02",
"A10BX03"])
        repaglinide          = Composante("repaglinide"       , u"Repaglinide",
["A10BX02"])
```

One of the most important aspects of the medication is represented by the *ATC (Anatomical Therapeutic Chemical)* codes ("A10BX02", "A10BX03") of the drugs. Based on these, the software allows the comparison between drugs and checks the appurtenance of drug into certain drug classes.

Treatment representation

Based on the definition of the drugs, then the treatment model is represented. An example of a treatment representation is presented in the next line:

```
        Traitement([mhd, glitazone], statut = PROPOSE, changement_de_dose = 0)
```

One can see that each treatment can have multiple medications associated ("mhd" and "glitazone" in the above example).

Medical Rules representation in ASTI 3

The final step is the representation of medical rules. Each medical rule has two main parts:

- the verification of different patient attributes (e.g., weight, age)
- the comparisons between proposed treatment and the current or past treatment

An example of Medical rules in *ASTI 3* [73] is presented beneath:

```
       if (((Traitement([mhd, antidiabetique_oral], statut = PROPOSE)) and (0.0 <
patient.HbA1c_max <= 6.5)) and
            (not Traitement([metformine, mhd], statut = PROPOSE)) and
            (Traitement([mhd, iag], statut = PROPOSE)) and
            not(
             Traitement([metformine, mhd], statut = EN_COURS_OU_PASSE, echec =
1) or
             Traitement([mhd, iag], statut = EN_COURS_OU_PASSE, echec = 1))
            ): moteur.critique(u"Les inhibiteur des alphaglucosidases ne sont
recommand\xe9s qu'en cas d'intol\xe9rance av\xe9r\xe9e \xe0 la metformine
(troubles digestifs) prescrite de fa\xe7on ad\xe9quate ou de contre-indication \xe0
cette mol\xe9cule (exceptionnelle dans un tel contexte ; si c'est le cas, votre
prescription est conforme).\nHbA1c entre 6% - 6.5% apr\xe8s 6 mois de mesures
hygi\xe9nodi\xe9t\xe9tiques : le groupe de travail recommande la prescription de
metformine.\n  (Traitement m\xe9dicamenteux du diab\xe8te de type 2 --
Recommandations page 20 -- 11/2006)", 3, 1) # patient 1, traitement 2
```

In order to identify the differences in the representation of medical rules, the first step was to represent manually in *Arden Syntax* the rules from *ASTI 3*. The major steps leading to this representation are identifying:

- which part of the rules are easy to translate, having almost the same syntax;
- how to declare the variables needed in the rules and how to retrieve their values for them;
- what extra functions should be defined and in which manner;
- how to represent drugs;
- how to represent treatments and;
- how to compare the different treatments.

Based on this I start the representation of different variables, functions or rules with *Arden2ByteCode* (implements *Arden Syntax*) [48] and *Egadss* open-source solution. For each part I tried to discover the most convenient representation, paying attention to the fact that each part should be implemented inside a rule and each part should be easy used in the rules in order to make the inference posible.

## 5.3. *ASTI 3* treatment model in *Arden Syntax*

In this section I try to identify if *Arden Syntax 2.8* is expressive enough to allow the representation of *ASTI 3* treatment model (by using the objects from *Arden Syntax*).

Starting from *Arden Syntax 2.5* object constructions can be defined. The object data type “may contain multiple named attributes, each of which may contain any valid *Arden* type (including lists or objects)” [27]. This feature of the standard may allow the object representation of the treatment model.

Because our objective is the representation of medical rules it means that our work will involve the *Knowledge* category of the *MLM*. It is not our concern to represent the *Maintenance* category, *Library* category and *Resource* category.

The *maintenance* category and *library* category are mandatory; they can be generated after the *knowledge* category is generated. The information which is contained by the different slots of these two categories can be completed based on the medical rules which are represented in the *knowledge* category, and can be done by a medical expert and/or an IT expert.

The representations of the different components of the ASTI with *Arden Syntax* are presented in the next paragraphs.

### 5.3.1. Evoking rules

In *ASTI 3* the rules are triggered when the physicians try to create (print) a medical recommendation. The *Arden Syntax* standardized language allows the representation of triggers based on events. The first step is the definition of an event variable. The usage and the definitions of the events are specific for each institution. In our case the definition of the event variable may be represented like:

```
event_rec= EVENT {new prescription action};
```

After the event variable is define it is used in the *evoke* slot. In the evoke slot, we have the next representation:

```
evoke:
        even_rec;;
```

This means in our representation, that when a physician tries a "*{new prescription action}*", the *MLM* containing the medical rules is evoked. Of course this implies extra mapping of the "{*new prescription action*}" to the local context through the help of the *Arden Syntax* execution engine. Although *Arden2ByteCode* and *Egadss* are the newest open-source solutions that implement *Arden Syntax*, they don't allow the execution of the *evoke* slot.

### 5.3.2. Special function representation and call.

As stated earlier, the *ASTI 3* implementation presents a number of functions used for different computations or evaluations. Regarding their representation in *Arden Syntax*, these functions can be represented in different *MLMs*, which are called from the main *MLM*, where the medical rules are defined. The *call* statement form the *Arden Syntax* allows the nesting of *MLMs*.

In the followings an example of function (represented in *ASTI 3*) for the computation of the *BMI* is presented:

```
def calcule_IMC(patient):
  if (patient.poids == 0) or (patient.taille == 0):
    print "* ASTI * ATTENTION: IMC non calculable (taille ou poids manquant) !"
    return 23.0
  return  10000.0 * patient.poids / (patient.taille ** 2)
```

***Arden Syntax* representation of special functions**

There are two steps in the nesting process. In the first step, from the main *MLM*, a call is done for the *MLM* in which the special function is represented. This call can be represented like this:

"(element1, element2) := call 'special_function_*MLM*' with first_parameter, second_parameter;"

We can see that the result of the special function is assigned to list containing *element1* (the *BMI* is assigned to it) and *element2* (the message is assigned if the *BMI* could not be calculated). Two parameters are sent to the second *MLM*, representing the weight and the size.

At this moment the main *MLM* is interrupted and the second *MLM* is executed. This *MLM* take the arguments which were specified in the main *MLM*, "first_parameter, second_parameter" (as presented in the example above). Based on the rules from the *logic* slot a result is returned to the main *MLM*.

A possible representation of this special function (in the second *MLM*) for the *Knowledge* slot in *Arden Syntax* is:

```
knowledge:
   type: data-driven;;
   data:
       (weight, size):=ARGUMENT;
           ;;
   evoke:
      ;;
   logic:
         mesaj:="";
         BMI:=23.0;
         if        (weight=0) or (size=0) then
         mesaj:="ATTENTION: BMI not calculable!";
         else BMI:=10000.0 * weight / (size ** 2);
         endif;
         conclude true;
   ;;
   action:
       return mesaj, BMI;
```

In order to avoid the use of objects in *Arden Syntax* (it is not allowed to pass objects as arguments between *MLMs* [27]) the values of the patient attributes (weight and size) are passed between the *MLMs*. We can observe that the simple logic from the *ASTI 3* can be, easy and in a similar way, represented in *Arden Syntax*.

### 5.3.3. Representation of the variables and connections to data sources

The rules are based on the comparison of:

- patient parameters (age, sex, HbA1c_max, HbA1c, size, weight) and
- treatments: the proposed treatment with the current treatment and/or the past treatment.

In *ASTI* the patient parameters are defined as in the following example:

```
        Critere("age"                              , "AGE")
```

In *Arden Syntax* they are defined in the *data* slot of the *Knowledge* category. They can be either extracted from a database or obtained from an interface. The link to the data sources (databases or visual interfaces) depends on the specifications from the curly brackets (example for retrieving from a database: "age:= read {select age from adatabase}"), and on the mapping of curly brackets to the local context through the help of the *Arden Syntax* execution engine. The retrieving of patient data depends on the translation of the information in curly brackets to a query engine.

In order to exemplify this, in the next sections, we will describe the implementation of the curly brackets in two well-known open-source software used for the execution of *MLMs* (*Egadss* and *Arden2ByteCode* [48]) and the use of *Arden Syntax* in collaboration with a *vMR* (*virtual Medical Record*).

***Egadss representation***

The first example is based on *Egadss* open-source software. In order to make the inference, *Egadss* use:

- medical rules represented in *Arden Syntax*;
- facts, which are extracted from a *HL7 CDA (Health Level 7 Clinical Document Architecture)* messages represented in a *XML* file.

A tool for mapping the information contained in curly brackets to a *XML* parser, is also implemented. When *Egadss* receives an *HL7 CDA XML*, the system extracts the information based on the specifications from the curly brackets; this information specifies the place in the *XML* where that patient information is present, as shown in the next lines:

```
let valPotassium be read
exist{num/ClinicalDocument/component/structuredBody/labsComponent/
section/code[@code="11502-
2"]/parent::section/entry/observation/code[@code="4548-4"]
/parent::observation/ value/attribute::value};
```

In this example we can see that the path in the curly brackets is used to specify the place where the potassium value can be found in the *HL7 CDA XML* [82].

***Arden2ByteCode representation***

In the second example we present the use of curly brackets in *Arden2ByteCode* software. Similar to *Egadss* open source, in order to make the inference, the medical rules are represented with *Arden Syntax*, in *MLMs*.

*Arden2ByteCode* allows the extraction of patient data by means of *SQL* queries. These queries are specified directly in the curly brackets. To make the queries, *Arden2ByteCode* enables the use of *JDBC* drivers [48] (beside this, the database location or the database type of has to be specified). A more complex description of this solution can be found in [48]. In the next line an example of curly brackets in *Arden2ByteCode*, is represented:

```
var_weight := read {select weight from person};
```

In this line, is assigned to the variable var_weight the value of the weight (needed for the rules execution) which is extracted from the person table. This variable can be used for the computation of the *BMI*.

***Connecting Arden Syntax execution engine to vMR***

The *CDS* can also be connected to a *vMR* [28]. An original way to do this, partially as contribution of the author, is described in [95]. In this paper, the representation of the *vMR* is made using *Topic Maps* technology and using *Topincs* software. A part of the *vMR* model containing age representation can be seen in Figure 5.3. The resulting information base is called *TM-vMR*. In order to integrate the *TM-vMR* with *Egadss* or *Arde2ByteCode* a translation of the patient data from the *vMR* to *HL7 CDA* format or to *SQL* databases can be made. For a direct connection to the *TM-vMR* by using the curly brackets, the paper [95] suggests an innovative approach, containing the next steps (the steps will be based on exemplifying the extraction of patient age):

a. Define of the curly brackets content. In our case the content of the curly brackets is:

   ```
   age := read {patient_ID, age};
   ```

b. The content of the curly brackets is extracted using parsing tools and passed through web services to a PHP function which interacts with the *TM-vMR.*
c. This function takes as arguments the ID of the patient and the needed data (age); the access to the different topics of the *vMR* is based on this data and is done through *Topic Maps* objects (*tobjects*). In the next line an example of how to work with *tobjects* is presented:

   ```
   $tobject1 = Tobject::get(ID);
   $age = $ tobject1 ->get_age();
   ```

   The way in which the *get_age* function is generated and which are the interactions between the different topics are described in [95].
d. The obtained age can then be passed to the *MLM*.

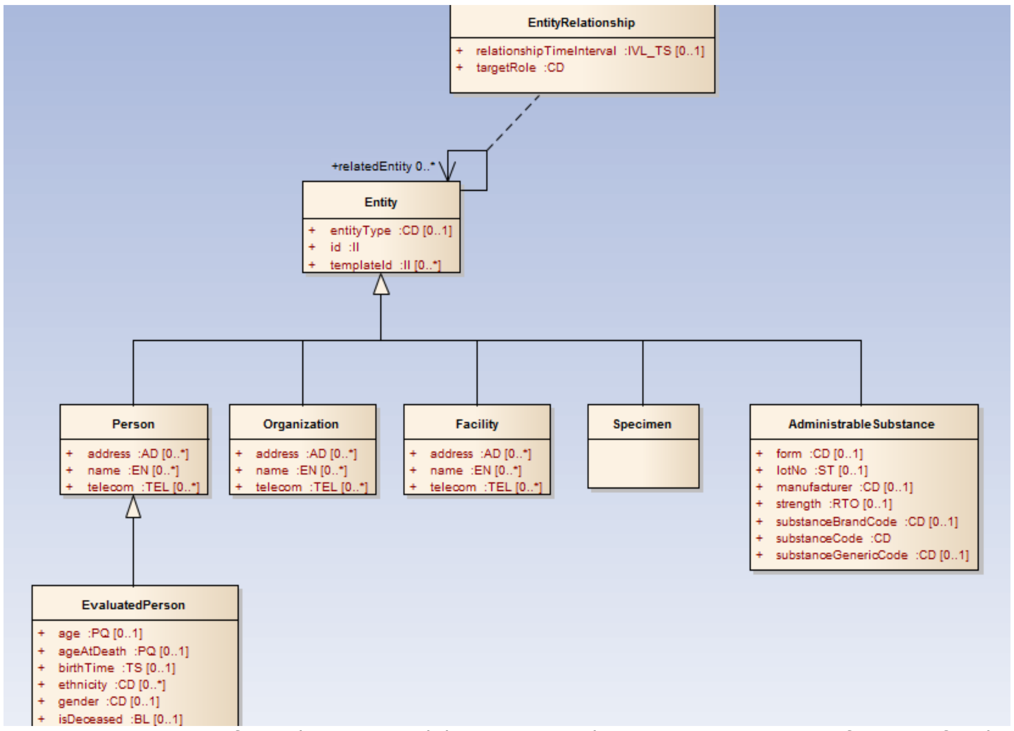


Figure 5.3. A section from the *vMR* model, containing the representation specifications for the person and other entities [28]

### 5.3.4. Medications representation

The medication representation in *ASTI 3* is made using objects that have different attributes: *ATC classification, codes, forme, dose, dose change, form change, tolerance.*

An instance of *DrugComponent,* part of the *Treatment* model in *ASTI 3*, is obtained like in the following example:

```
sulfamide  = DrugComponent("sulfamide", u"Sulfamide", ["A10BB"])
```

*Arden Syntax* allows similar constructions for the representation of this model part. First an object type must be define:

```
medication := object [classification_name, drug_name, codes, form, dose,
dose_change, form_change, tolerance] ;
```

Afterwards an instance of this object can be created:

```
medication1 := new medication with "val_durg_name", "val_codes", "val_form",
"val_dose", "val_dose_change", "val_form_change", "val_tolerance"
```

The attributes, contained by the medication object, are associated with the next data types:

- drug_name list of Strings;
- codes is List;
- forme is String;
- dose is String;
- dose_change is String;
- form_change is String;
- tolerance is String;

We can see that for each attributes the data type is Strings, except codes. The strings are needed for the representation of "*any*" value – from the model. In order to make some computational operations the next conversions may be needed:

- Dose is converted from String to Number if the value is different from "any", "known" and "not known";
- Dose change is converted from String to Boolean if the value is different of "any";
- Form change is converted from String to Boolean if the value is different of "any".

The representation of different *ACT* codes in the medication representation is another special topic. In *ASTI 3* language for a medication there multiple *ACT* codes can be defined (e.g.,*"insulino_secreteur = Composante("insulino_secreteur" , u"Insulino-secreteur",* ***["A10BB", "A10BX02", "A10BX03"]****)"*). In *Arden Syntax* the representation of this part can be achieved through the use of a List. In the *if–then* rules the elements of different lists (of different medication) need to be compared to see if a medication can be prescribed, a more complex presentation of this can be found in section 5.6.

### 5.3.5. Treatment Representation

The treatment representation is one of the most complex parts of *ASTI 3* model. It is based on the representation of medications.

A treatment in *ASTI 3* is composed of: a List medication objects (see section 4.4), *status, seniority, efficacy, dose change, form change, INN change*. In *ASTI 3* this is represented with the help of objects. In *Arden Syntax* the representation can be similar and is also realized with the help of objects.

An instance of treatments model in *ASTI 3*:

```
Traitement([mhd, insuline_simple, glinide, iag          ], statut = PROPOSE,
changement_de_dose = 0)
```

*Arden Syntax* allows similar constructions for the representation of *Treatment* model. First an object type must be defined:

```
treatment := object [list_medication, indication, status, seniority, efficacy,
doseschange, fromchange, innchange] ;
```

Then, an instance of this object can be created like:

```
Treatment1 := new treatment with list_medication, “val_indication”, “val_status”,
“val_seniority”, “val_efficacy”, “val_doseschange”, “val_fromchange”,
“val_innchange”
```

The attributes, contained by the treatment object, are associated with the next data types:

- List of medicationobjects;
- Indication is a String;
- Status is a String;
- Seniority is a String;
- Efficacy is a String;
- Dose change is a String;
- From change is a String;
- INN change is a String.

Similar for the medication representation, we can see that for each attributes the data type is String, except for medication, which is a List of objects. The Strings are needed for the representation of: “*any*”, “*past or current*”, “*current or proposed*”, “*partial or null*”– from the model. In order to make some computational operations the next conversions may be needed:

- Seniority is converted from String to Number, if the string value is different from “*any*”;
- Efficacy is converted from String to Number if the value is different from “*partial or null*” and “*any*”;
- Dose change is converted from String to Boolean if the value is different from “*any*”;
- Form change is converted from String to Boolean if the value is different from “*any*”;
- INN is converted from String to Boolean if the value is different from “*any*”.

In the representation of the *Treatment* model, another problem is constituted by the implementation of the class *Any*. One possible representation of this can be achieved by defining an object with only one String attribute:

```
        Object_any := object [any_information]
```

There can be one instance of this object or none. The object can be used for the representation of all others medications which are not defined. Its use and role in the medical rules will be presented in the next section.

### 5.3.6. Rules representation

The rule representation is the main aspect of this analysis. The representation of medical rules involves the comparison of different: medications, patient data or treatment attributes.

In *ASTI 3* medical rules are represented in the following manner:

```
        if (((Traitement([mhd, antidiabetique_oral], statut = PROPOSE)) and
((patient.HbA1c_max == 0.0) or (patient.HbA1c_max > 6.5)) and (patient.BMI <
27)) and
           (not Traitement([metformine, mhd], statut = PROPOSE)) and
           (not Traitement([mhd, sulfamide], statut = PROPOSE)) and
           (not Traitement([glinide, mhd], statut = PROPOSE)) and
           (Traitement([Any], statut = PROPOSE)) and
           not(
            Traitement([metformine, mhd], statut = EN_COURS_OU_PASSE, echec =
1) or
            Traitement([mhd, sulfamide], statut = EN_COURS_OU_PASSE, echec = 1)
or
            Traitement([glinide, mhd], statut = EN_COURS_OU_PASSE, echec = 1))
           ): moteur.critique(u"critique, explanation, reference", 2, 2) # patient 1,
traitement 2
```

In this section we will try to represent (in *Arden Syntax*) each condition of the above medical rule and try to identify all extra functions that should be created in order to make the different comparisons. For this representation we assume that the different attributes used in the rules (medications, treatment, patient data) were already represented and values were assigned, as presented in the previous sections (5.3, 5.4 and 5.5). We will try to find out if the *Arden Syntax 2.8* has enough expressiveness in order to translate the *ASTI 3* rules based on the previously analyzed concepts.

For this rules we identified two main parts:

- a part used for comparing patient data:

```
((patient.HbA1c_max == 0.0) or (patient.HbA1c_max > 6.5))
```

- a part where different treatments are compared:

```
(not Traitement([glinide, mhd], statut = PROPOSE))
```

The representation of the part based on the patient data is an easy task, because *Arden Syntax* offers all the comparison operators needed:

```
       ((HbA1c_max=0.0) or (HbA1c_max>6.5))
```

Analyzing the "not Treatment(…)" statement of the of the *ASTI 3* rules, we can see that these is no direct translation of such use case of objects in *Arden Syntax*. For instance in *Arden Syntax* we have the comparison *operators is [not] object* and *is [not] <Object-Type>*, which are true if a certain variable is an object or if the variable is an object type previously defined [27]. In order to represent just the (not Traitement([glinide, mhd], statut = PROPOSE)) condition part of the rule, this implies multiple verification in *Arden Syntax*.

### *Treatments comparison*

In this subsection the representation of the *Knowledge* category of a *MLM* is exemplified. The rules contain two types of conditions: regarding the treatment and another on the patient data.

For this we propose for translation a simplified rule:

```
	if (((patient.HbA1c_max == 0.0) or (patient.HbA1c_max > 6.5)) and (not
Traitement([glinide, mhd], statut = PROPOSE))): moteur.critique(u"critique,
explanation, reference", 2, 2) # patient 1, traitement 2
```

In *ASTI 3*, the not Traitement([glinide, mhd], statut = PROPOSE) statement means that, if the treatment proposed by the physician is not containing glinide and mhd, the statement is true. In order to represent this and the conditions on the patient data, multiple steps need to be performed:

a. Definition of the drug and treatment object:

```
drug := object[atc_classification, codes_list, forme, dose, dose_change,
form_change, tolerance] ;
treatment := object[drug_list, indication, status, seniority, efficacy, doseschange,
fromchange, innchange] ;
```

b. Creating the instance for *glinide* and *mhd* (the current medication, to which the prescribed medications, from the proposed treatment are compared):

```
drug1 := new drug;
drug1.atc_classification:="mhd";
drug1.codes_list:= "P0026";
drug2 := new drug;
drug2.atc_classification:="glinide";
drug2.codes_list:= "A10BX02", "A10BX03";
```

c. Reading the values that should be compared (treatment and patient data), for this we suppose that the physician is trying to propose a treatment with 2 medication:

```
HbA1c_max:=read {select HbA1c_max from person};
read_ treatment:= new treatment;
read_drug1 := new drug;
read_drug1.atc_classification:= {select atc_classification from interface};
read_drug1.codes_list:= { select atc_classification from drugDB where
atc_classification= read_drug1.atc_classification };;
read_drug2 := new drug;
read_drug2.atc_classification:= {select atc_classification from interface};
read_drug2.codes_list:= {select atc_classification from drugDB where
atc_classification= read_drug2.atc_classification };;
read_ treatment.status:= {select treatmentstatus from interface};
read_ treatment.drug_list:= read_drug1, read_drug2;
```

d. Definition of other variables needed for the comparison:

```
verification1=false;
mesaj:=””;
verification2=false;
```

e. Building the comparison:

```
for j in (1 seqto count atraitment.drug_list) do
for h in (1 seqto count atraitment.drug_list[j].codes_list) do
for m in (1 seqto count drug1.codes_list) do
if (FIND read_ traitment.drug_list[j].codes_list[h] IN STRING drug1.codes_list[m]
STARTING AT 1 <> 0)  then verification1:=true;
endif;
enddo;
for m in (1 seqto count drug2.codes_list) do
if (FIND read_ traitment.drug_list[j].codes_list[h] IN STRING drug2.codes_list[m]
STARTING AT 1 <> 0)  then verification2:=true;
endif;
enddo;
enddo;
enddo;
```

By using for loops we compare if the medication proposed by the physician is mhd or glinide.

f. The rule obtained:

```
if (verification1 and verification2 and (read_ traitment.status="PROPOSE")) then
mesaj :=" critique, explanation, reference"
endif;
conclude true;
```

## 5.4. Problems raised by the objects representing *ASTI 3* rules in *Arden Syntax* version 2.8

In the previous sub-chapter I analyzed how *ASTI 3* rules can be expressed in *Arden Syntax* version 2.8. For this representation I used the objects from *Arden Syntax* formalism.

There are several problems regarding the modeling of *ASTI 3* rules in *Arden Syntax*:

a. How the model can be modified in order to represent treatments in an easier way in *Arden Syntax* without the use of objects. Although the objects are defined starting with version 2.5 of *Arden Syntax,* their use is not recommended because they are hardly understood by physicians.
b. How to extract information from data sources in order to avoid the use of constructions like objects containing lists containing other objects. These constructs are not allowed in most *Arden Syntax* execution environments.

In the next sections all the previous problems will be further discussed and several solutions are presented.

## 5.5. Representation of *ASTI 3* medical rules using multiple *MLMs*

Based on discussion with Dr. Harry Karadimas[1] regarding the previous representation of *ASTI 3* rules in *Arden Syntax*, two major conclusions emerged:

- regarding the *Arden Syntax 2.8* standard the representation is correct;
- regarding the use of objects in *Arden Syntax*, constructions like object.list.list (e.g., "read_ traitment.drug_list[j].codes_list[h]") are not recommended in the development of *MLMs* (because one of the main characteristic of *Arden Syntax* should be it simplicity, in order to be easy understood by medical experts).

Based on this conclusion a second representation of the medical rules is proposed in order to avoid the use of complex constructions (e.g., "read_ traitment.drug_list[j].codes_list[h]"). To realize this I propose a representation where multiple *MLMs* are used.

For the implementation of the rules we identified 3 *MLM* types beside the ones used for the representation on special functions like *BMI* computing.

### 5.5.1. First *MLM* representation

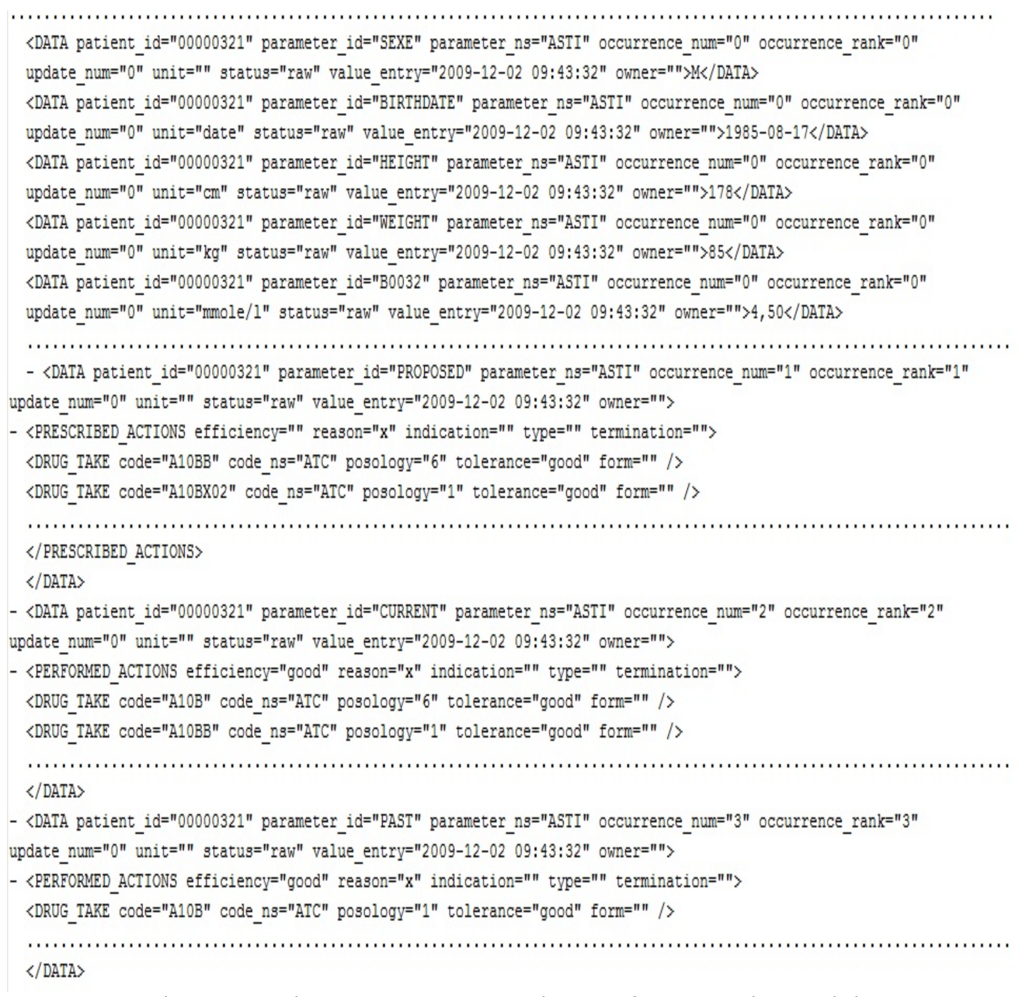
The first *MLM* is used for the representation of simple condition (verification of *BMI* or the verification of the treatment *status*) and call a second *MLM* for the verification of drugs based on the attributes of a specific treatment

I use the next *ASTI* rule for exemplification in *Arden Syntax* (for all three *MLMs* types):

```
 if ((Traitement([mhd, antidiabetique_oral], statut = PROPOSE)) and (not
Traitement([metformine, mhd], statut = PROPOSE)
```

- The input of the system is an *XML* file containing the patient data (e.g., age, sex, weight) and the treatments associated with that patient (Figure 5.4).

[1] Dr *Harry Karadimas*, researcher in the Department of Information of *Henri Mondor University Hospital, France*. Specialized in clinical decision support systems and *Arden Syntax* formalism.



```
..........................................................................................................
  <DATA patient_id="00000321" parameter_id="SEXE" parameter_ns="ASTI" occurrence_num="0" occurrence_rank="0"
  update_num="0" unit="" status="raw" value_entry="2009-12-02 09:43:32" owner="">M</DATA>
  <DATA patient_id="00000321" parameter_id="BIRTHDATE" parameter_ns="ASTI" occurrence_num="0" occurrence_rank="0"
  update_num="0" unit="date" status="raw" value_entry="2009-12-02 09:43:32" owner="">1985-08-17</DATA>
  <DATA patient_id="00000321" parameter_id="HEIGHT" parameter_ns="ASTI" occurrence_num="0" occurrence_rank="0"
  update_num="0" unit="cm" status="raw" value_entry="2009-12-02 09:43:32" owner="">178</DATA>
  <DATA patient_id="00000321" parameter_id="WEIGHT" parameter_ns="ASTI" occurrence_num="0" occurrence_rank="0"
  update_num="0" unit="kg" status="raw" value_entry="2009-12-02 09:43:32" owner="">85</DATA>
  <DATA patient_id="00000321" parameter_id="B0032" parameter_ns="ASTI" occurrence_num="0" occurrence_rank="0"
  update_num="0" unit="mmole/l" status="raw" value_entry="2009-12-02 09:43:32" owner="">4,50</DATA>
  ..........................................................................................................
  - <DATA patient_id="00000321" parameter_id="PROPOSED" parameter_ns="ASTI" occurrence_num="1" occurrence_rank="1"
update_num="0" unit="" status="raw" value_entry="2009-12-02 09:43:32" owner="">
- <PRESCRIBED_ACTIONS efficiency="" reason="x" indication="" type="" termination="">
  <DRUG_TAKE code="A10BB" code_ns="ATC" posology="6" tolerance="good" form="" />
  <DRUG_TAKE code="A10BX02" code_ns="ATC" posology="1" tolerance="good" form="" />
  ..........................................................................................................
  </PRESCRIBED_ACTIONS>
  </DATA>
- <DATA patient_id="00000321" parameter_id="CURRENT" parameter_ns="ASTI" occurrence_num="2" occurrence_rank="2"
update_num="0" unit="" status="raw" value_entry="2009-12-02 09:43:32" owner="">
- <PERFORMED_ACTIONS efficiency="good" reason="x" indication="" type="" termination="">
  <DRUG_TAKE code="A10B" code_ns="ATC" posology="6" tolerance="good" form="" />
  <DRUG_TAKE code="A10BB" code_ns="ATC" posology="1" tolerance="good" form="" />
  ..........................................................................................................
  </DATA>
- <DATA patient_id="00000321" parameter_id="PAST" parameter_ns="ASTI" occurrence_num="3" occurrence_rank="3"
update_num="0" unit="" status="raw" value_entry="2009-12-02 09:43:32" owner="">
- <PERFORMED_ACTIONS efficiency="good" reason="x" indication="" type="" termination="">
  <DRUG_TAKE code="A10B" code_ns="ATC" posology="1" tolerance="good" form="" />
  ..........................................................................................................
  </DATA>
```

Figure 5.4. The *XML* used in *ASTI 3* containing the specific patient data and the treatments associated

The main steps in the representation of the *MLM* are: retrieving treatment parameters and the logic slot representation. These steps are described in the next paragraphs.

***Retrieving treatment parameters***

Retrieving of information in *Arden Syntax* is made based on the specification from the curly brackets. The text from the brackets is analyzed by the execution engine and, the required data is retrieved.

```
knowledge:
   type: data-driven;;
   data:
     n := read {extract numbe_of_tretments from XML };
```

- Based on the mapping of the curly brackets to the local context through the execution engine, the read statement should return the number of treatments from the *XML* (containing the treatments of a certain patient).

```
status:=read {extract treatments_status from XML};
```

- Based on the mapping of the curly brackets to the local context through the execution engine, the read statement should return a list of all statuses. If for a treatment no status is found *null* will be returned.

```
seniority:=read {extract treatments_seniority from XML};
```

- Based on the mapping of the curly brackets to the local context trough the execution engine, the read statement return a list of all seniority. If for a treatment no seniority is found *null* will be returned.

```
dose_change:=read {extract treatments_dose_change from XML};
```

- Based on the mapping of the curly brackets to the local context through the execution engine, the read statement return a list of all donse change if for a treatment no dose change is found *null* will be returned.

```
form_change:=read {extract treatments_form_change from XML};
```

- Based on the mapping of the curly brackets to the local context through the execution engine, the read statement return a list of all form change if for a treatment no form change is found *null* will be returned.

```
efficacy := read {extract efficacity from XML};
```

- Based on the mapping of the curly brackets to the local context trough the execution engine, the read statement return a list of all efficacy, if for a treatment no efficacy is found, *null* will be returned.

***Arden Syntax logic slot representation***

```
logic:
for i in (1 seqto n) do
If (status[i]="Propose") then
```

- Beside the condition which regards the status of the treatment, other condition can be added for other attributes of the treatments if this is specified in the *ASTI* rule. In this specific case, for each treatment from a list of treatments it is verified if its status is equal to "Propose". The "i" variable represent the number of the treatment in the treatments list.

```
element1 := call secondMLM with mhd, i;
element2 := call secondMLM with antidiabetique_oral, i;
```

- The two values, element1 and element2 represent boolean values and is true or false if the drug is present in a treatment. The value of the two attributes is computed in the "second*MLM*" based on the two arguments: drug ("antidiabetique_oral") and treatment identification. The execution of the first *MLM* is interrupted when the second is called.

```
if element1 and element2 then c1:=true;
else c1:=fase;
endif;
```

- This "if-then" rule is used to verify if both medications are part of the treatment (in our case "mhd" and "antidiabetique_oral") based on the returned values from the second *MLM*.

```
If (status[i]="Propose") then element1 := call secondMLM with mhd, i;
element2 := call secondMLM with metformine, i;
if element1 and element2 then c2:=true;
else c2:=fase;
endif;
endif;
```

- This *if* rule is equivalent to the next part of the *ASTI 3* rule: "(not Traitement([metformine, mhd], statut = PROPOSE)".

```
enddo;
if c1 and (not c2) then conclude true;
endif;
      ;;
action:
write "critism, eplanation, references" ;
;;
end:
```

- Retuned message: if the rule is true then the *MLM* will return a message containing *critisism, eplanation* and *references*.

### 5.5.2. Second *MLM* representation

In the second *MLM* all the drugs of a treatment are identified. Based on this drugs the third *MLM* is called (a description of this *MLM* is presented in section 5.8.3).

```
knowledge:
   type: data-driven;;
   data:
        (drug, position_of_treatment):=ARGUMENT;
    n := read {extract numbe_of_drugs from XML for treatment at
position_of_treatment};
;;
```

- Based on the mapping of the curly brackets to the local context through the execution engine, the read statement should return the number of medications for the specified treatment.

```
evoke:
      ;;
logic:
for i in (1 seqto n) do
element1 := call thirdMLM with mhd, position_of_treatment, i;
enddo;
conclude true;
      ;;
```

- Here the third *MLM* is called, it extract a drug and compares it with the drug that is sent as parameter, in our case "mhd". The other parameters which are sent: the position of the treatment and the position (variable "i" in our example) of the drug for that treatment

```
action:
return element1;
;;
end:
```

- The result of the comparison is returned to the first *MLM*

### 5.5.3. Third *MLM* representation

The third *MLM* contains the actual comparison between the drugs specified in the "*if-then*" rule and the drugs from the treatment

```
knowledge:
   type: data-driven;;
   data:
     (drug_for_verification, position_of_treatment, position_of_drug):=ARGUMENT;
     list_codes_from_treatment := read { extract drug_codes from treatment at
position_of_treatmentXML with drog at position_of_drug};
```

- From the second *MLM* three arguments are passed: drug to be verified ("drug_for_verification"), position of the treatment in the list of treatments ("position_of_treatment") and position of the drug in the list of drugs ("position_of_drug").

```
     list_codes_for_drug := read { extract drug_codes from drugDB where drug is
```

```
drug_for_verification};
     ;;
```

- Based on the mapping of the curly brackets to the local context trough the execution engine, the read statement should return a predefined list of drug codes for the drug which is used in the medical rule
- In this example we compare drugs based on the *ATC* codes

```
evoke:
       ;;
logic:
verification:=false;
for j in (1 seqto count list_codes_for_drug) do
for h in (1 seqto count list_codes_from_treatment) do
if ((FIND list_codes_for_drug IN STRING list_codes_from_treatment STARTING AT
1) <> 0)  then verification:=true;
endif;
 enddo;
 enddo;
```

- Based on the *ATC* code a verification if the drug code from the treatment is part part of the code drug from the "list_codes_for_drug"

```
conclude true;
       ;;
   action:
return verification;
;;
end:
```

- The result of the evaluation is returned to the second *MLM*.

In this implementation we can see that the use of the objects was completely avoided by using multiple *MLMs*.

## 5.6. Implementation of *ASTI 3* rules based on external functions.

A third method for the representation of *ASTI 3* rules is based on external functions. These functions are used to verify if a drug is part of a certain treatment and they are called through the curly brackets.

In order to implement "Traitement([mhd, antidiabetique_oral], statut = PROPOSE)" condition from *ASTI 3* rules, in the *data* slot of the *Knowledge* category of an *MLM* we have:

```
antidiabetique_oral:= "A10B";
        mhd := "P0026";
        let drug1 be read exist {select drug from treatment where
drug=antidiabetique_oral and status="propose"};
```

```
        let drug2 be read exist {select drug from treatment where drug=mhd and
status="propose"};
```

- Verification if the drug is antidiabetique_oral, comparing for example, the *ATC* code is done by the execution engine and through an external function.

In the *logic* slot we have:

```
if exists drug1 and  exists drug2 then conclude true;
```

In order to test the *MLM* files which were translated from *ASTI 3* we chose the *vMR* model for the representation of patient data (because it is the only standard intended for the representation of patient data for clinical decision support systems). In order to use the *vMR* an analysis of how the data needed for the inference in *ASTI 3* can be found in the model, needs to be done.

## 5.7. Adaptation of *ASTI 3* model to *vMR* model

We will make an analysis to identify the components of *ASTI 3* treatment model in the *vMR* model. This analyze is done in order to see if the information needed by *ASTI 3* (to make the inference) can be represented in *vMR* standard format.

We will see also from where (in the *vMR*) can the patient data be extracted in order to specify proper queries in the curly brackets of *Arden Syntax*. Is not the purpose of this work specifying the queries from the curly brackets but is useful to identify the data elements needed for the inference based on the *MLMs* containing the *ASTI 3* medical rules. In this way after representing the rules from *ASTI 3* in *Arden Syntax* we also identify the representation of the patient data from the *ASTI 3* model into the *vMR*.

### 5.7.1. Which data from ASTI 3 model can be mapped to *vMR*

Studying the data needed for the inference in both *ASTI 3* and the *vMR* model I have identified the elements which can be mapped and which cannot be mapped from *ASTI 3* to *vMR*. In table 5.1 we can see how the elements from the *ASTI 3* data model can be represented in *vMR* model.

| ***ASTI* Class** | ***ASTI* data element** | **Types in ASTI** | ***vMR* Class** | ***vMR* Data** | **Types in *vMR*** |
|---|---|---|---|---|---|
| DrugComponent | clasification | String (ATC) | AdministrableSubstance | It can be found in substance code (The code that identifies the substance with as much specificity as appropriate, or as required by a template. E.g., aspirin, lisinopril. May be either a generic or brand code, unless otherwise restricted by a template.) and is the | CD |

| | | | | | |
|---|---|---|---|---|---|
| | | | | systremCode from CD | |
| DrugComponent | codes | List of string | AdministrableSubstance | It can be found in substance code (The code that identifies the substance with as much specificity as appropriate, or as required by a template. E.g., aspirin, lisinopril. May be either a generic or brand code, unless otherwise restricted by a template.) and is the code from CD | CD |
| DrugComponent | forme | String (tablet) | AdministrableSubstance | form (The physical form of the substance as presented to the subject. E.g., tablet, patch, injectable.) | CD |
| DrugComponent | dose | float | SubstanceAdministrationBase | doseQuantity (The amount of substance. E.g., 1 tab, 325 mg, 1-2 tabs.) | IVL_PQ |
| DrugComponent | dose change | boolean | | No direct representation<br>This part can be calculated by comparing the previous drug dose with the one prescribed | |
| DrugComponent | form change | boolean | | No direct representation<br>This part can be calculated by comparing the previous drug form with the one prescribed. | |
| DrugComponent | tolerance | string | AdverseEvent | Severity (The intensity of the adverse event. E.g., severe, moderate. If the adverseEventCode is rash and severity is moderate, it means that the adverse event was a moderate rash.)<br>If no adverse event can be found (for a medication, adverseEventAgent ) then the tolerance is good – in *Arden Syntax*, in order to retrieve the tolerance the interrogation might look like: {extract severity of the} | CD |
| Treatment Patern | status | string | | No direct representation | |
| Treatment Patern | seniority | int | | No direct representation | |
| Treatment Patern | efficacy | string | | No direct representation | |
| Treatment Patern | dose change | boolean | | No direct representation | |
| Treatment Patern | form change | boolean | | No direct representation | |
| Treatment Patern | INN change | boolean | | No direct representation | |
| Treatment Patern | Indication | string | | No direct representation | |

Table 5.1. *ASTI* model mapped to *vMR*

There is no direct representation of the treatment components in *vMR* data model. Consequently we propose an extension of the *vMR* model in order to integrate all *ASTI* treatment model components.

### 5.7.2. New data elements that need to be considered in the model extension

As presented in table 5.1, we cannot find a representation in the *vMR* data model for any attributes of the *TreatmentPatern* class. In order to propose new components an analysis was made to identify the data types used and the relation of the *vMR* with *HL7 RIM (Reference Information Model)*. We can see that there are classes which are not present in *RIM*.

In the development of the new components the first step is to decide which information is mandatory in the extension of the *vMR* model by analyzing the data contained in the *ASTI 3* model and the approach for representing this data into the model.

In the next paragraph the attributes which could not be direct found in the *vMR* are enumerated, along with a short analysis of their importance and necessity to integrating them in the *vMR* model.

A. *Indication* attribute

This attribute refers to the indication which is associated with a treatment (set of drugs). The information regarding a substance administration can already be found in the *vMR* model. The attribute from *vMR* which can be used to specify this is *substanceAdministrationGeneralPurpose,* or attribute of the *SubstanceAdministrationBase* class. In order to implement this *ASTI 3* attribute, adding to the model a new attribute which describe the *indication* of a set of drugs may be needed.

B. *Status* attribute

In the *ASTI 3* model we can see that the status attributes can be: proposed, current, past, past or current, current or proposed, any – for each treatment. This values are used to express the status of a treatment for a certain patient. There can be multiple past treatments, which were replaced due to different issues (e.g., inefficiency).

The question which rises regarding the representation of these statuses in *vMR*, is whether they can be expressed for each drug separately or if new classes or new attributes through which to express the status, are needed.

Analysis revealed two possibilities:

- to express the proposed status we can use the *SubstanceAdministrationProposal* – class of the *vMR*,
- to express the current and past statuses we can use the *SubstanceAdministrationEvent* – class of the *vMR* - by using the *administrationTimeInterval* attribute, which allows the specification of a low time interval (for the beginning of the administration) and a high time interval (for the end of the administration). If there is no

"high time interval" specified for the *administrationTimeInterval* attribute this means that the administration of the drug is current. For the past drug administration the high time interval is mentioned.

In order to specify whether a treatment is current, past or proposed we can specify this status for each drug.

Regarding this approach, the biggest concern is the need of extra functions for the classification of different drugs (based on the use of different classes or attributes) in *past*, *current* or *proposed* status. Beside this classification, the past drug administrations have to be classified, based on the date, for different treatments (ex. Treament1 starting at date1, ending at date2).

C. *Seniority* attribute

This attribute is used to specify the order in which different treatments were administrated.

This can be computed based on the classifications made for the statuses of the drug administration (described in the previous section).

D. *Efficacy* attribute

This attribute refers to the effectiveness of a treatment (set of drugs). The efficacy of a substance administration cannot be found in the *vMR* model. In order to implement this *ASTI 3* attribute, adding to the *vMR* model a new attribute which describes the efficacy of a set of drugs is mandatory.

E. *Dose change* attribute

This attribute specify if there was a dose change for a drug. This information can be obtained by verifying if for a drug which is part of a treatment the dose changed compared whit the dose of the same drug from the previous treatment.

As a disadvantage extra function need to be created in order to compute this.

F. *Form change* attribute

This attribute specifies whether there was a form change for a drug. This information can be obtained by verifying if for a drug which is part of a treatment the form changed compared with the form of the same drug from the previous treatment.

As a disadvantage extra function need to be created in order to compute these values.

G. *INN Change* attribute

This attribute specifies if there was an *INN* (International Nonproprietary Names) change for a drug. This information can be obtained by verifying if for a drug which is part of a treatment the *INN* changed compared with the *INN* of the same drug from the previous treatment.

As a disadvantage extra function need to be created in order to compute these values.

Concerning the above mentioned challenges and statements regarding the representation of different *ASTI 3* attributes, we propose the next extension to the *vMR* model (Figure 5.5).

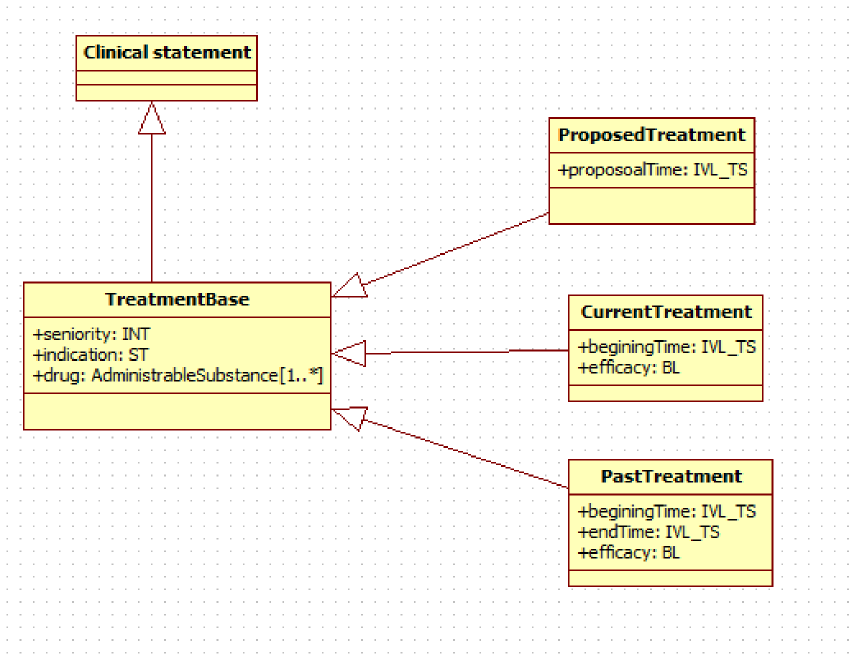


Figure 5.5. TreatmentBase for *vMR* model

In the Figure 5.5, the UML representation of the different new elements to be added to the *vMR*, is presented. The following four new classes are needed:

- TreatmentBase, which inherits Clinical statement class, and has as attributes: seniority, indication and drug;
- ProposedTreatment, which inherits TreatmentBase class, and has as own attributes: proposalTime (which is the time the treatment was proposed, indication and drug);
- CurrentTreatment, which inherits Tream*TM*entBase class, and has as own attributes: beginingTime (which is the time the treatment begin, indication and drug;) and efficacy;

PastTreatment, which inherits Tream*TM*entBase class, and has as own attributes: beginingTime, endTime, efficacy.

## 5.8. Testing and evaluation of the proposed translation

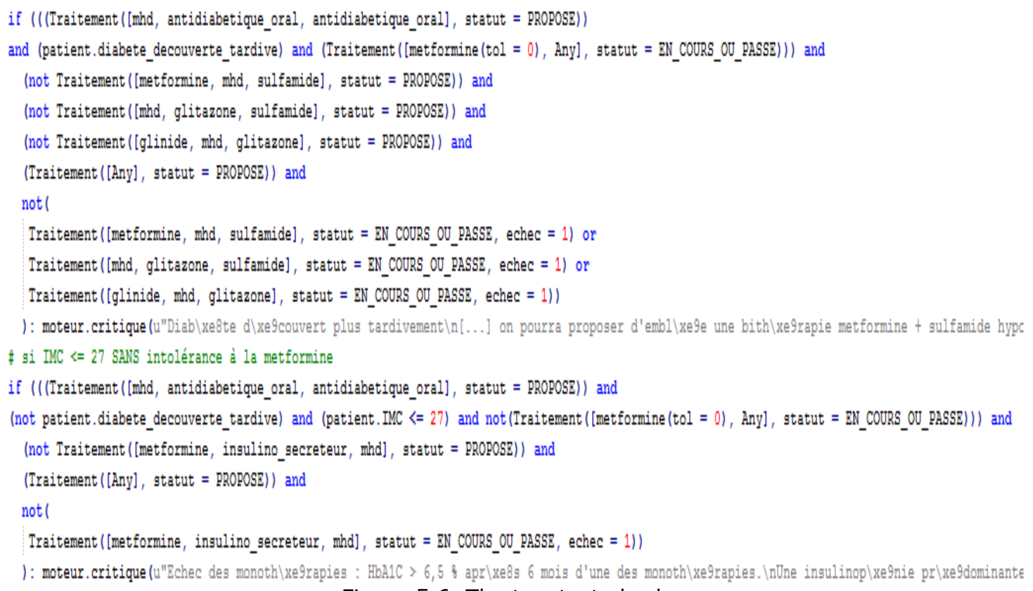
In order to test the first method of translating *ASTI 3* to *Arden Syntax* (using object representation in *Arden Syntax*), there were open-source software solution which implement list of constructions like this: "traitment.drug_list[j].codes_list". *Arden2ByteCode* use objects but not this type of construction which involve using "list of object of object" and *Egadss* doesn't allow object construction in *Arden Syntax* because it implements an older version (version 2.0) [16].

For testing the implementation using 3 different *MLMs* for the representation of medical rules, *Arden2ByteCode* and the system presented in Chapter 3 were used.

The next steps were taken for the evaluation:

- Translation of rules from *ASTI* to *Arden Syntax*;
- Representation of medical data (from the *Timişoara Emergency County Clinical Hospital*.) with the help of the *TM-vMR*;
- Generating recommendations based on the rules and data from *TM-vMR*;
- Comparing the recommendations with the one based on *ASTI.*

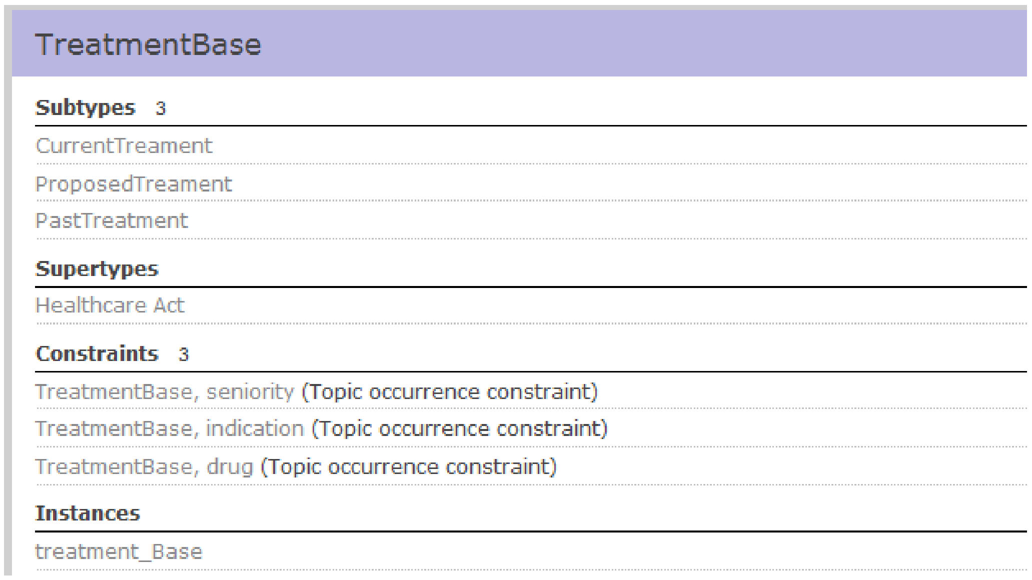
The testing method was applied for 20 different rules (In Figure 5.6, 2 such rules are presented) in *ASTI* representation using medical data from the *Timişoara Emergency County Clinical Hospital*. For each set of data the generated recommendations were the expected ones, being the same as the ones generated based on *ASTI 3* rules (from which the *MLM*, were translated). *ASTI* 3 rules which were used, represent critiques for physician if a prescribed treatment if not conform to the guideline.



```
if (((Traitement([mhd, antidiabetique_oral, antidiabetique_oral], statut = PROPOSE))
and (patient.diabete_decouverte_tardive) and (Traitement([metformine(tol = 0), Any], statut = EN_COURS_OU_PASSE))) and
  (not Traitement([metformine, mhd, sulfamide], statut = PROPOSE)) and
  (not Traitement([mhd, glitazone, sulfamide], statut = PROPOSE)) and
  (not Traitement([glinide, mhd, glitazone], statut = PROPOSE)) and
  (Traitement([Any], statut = PROPOSE)) and
  not(
   Traitement([metformine, mhd, sulfamide], statut = EN_COURS_OU_PASSE, echec = 1) or
   Traitement([mhd, glitazone, sulfamide], statut = EN_COURS_OU_PASSE, echec = 1) or
   Traitement([glinide, mhd, glitazone], statut = EN_COURS_OU_PASSE, echec = 1))
  ): moteur.critique(u"Diab\xe8te d\xe9couvert plus tardivement\n[...] on pourra proposer d'embl\xe9e une bith\xe9rapie metformine + sulfamide hypc
# si IMC <= 27 SANS intolérance à la metformine
if (((Traitement([mhd, antidiabetique_oral, antidiabetique_oral], statut = PROPOSE)) and
(not patient.diabete_decouverte_tardive) and (patient.IMC <= 27) and not(Traitement([metformine(tol = 0), Any], statut = EN_COURS_OU_PASSE))) and
  (not Traitement([metformine, insulino_secreteur, mhd], statut = PROPOSE)) and
  (Traitement([Any], statut = PROPOSE)) and
  not(
   Traitement([metformine, insulino_secreteur, mhd], statut = EN_COURS_OU_PASSE, echec = 1))
  ): moteur.critique(u"Echec des monoth\xe9rapies : HbA1C > 6,5 % apr\xe8s 6 mois d'une des monoth\xe9rapies.\nUne insulinop\xe9nie pr\xe9dominante
```

Figure 5.6. The two tested rules

In order to complete the test the new proposed components of the *vMR* model where represented by using the *TM-vMR* data base (see section 3.4). The needed data was retrieved from this source based on the *Data Manager* module and the *GetData* web service (which were developed using Windows Communication Foundation [122]). The TreatmentBase representation in *TM-vMR* can be seen in figure 5.7.



TreatmentBase

**Subtypes** 3
CurrentTreament
ProposedTreament
PastTreatment

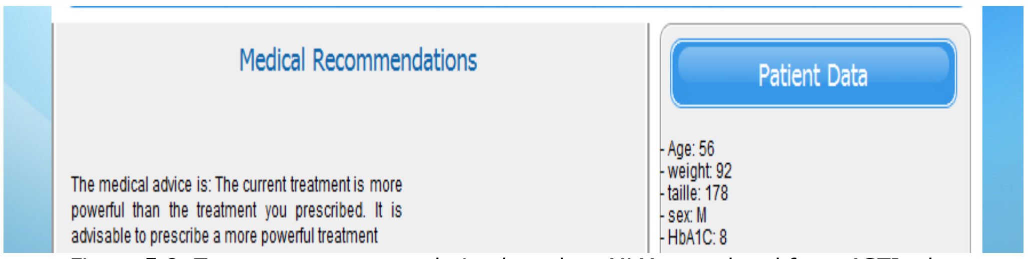
**Supertypes**
Healthcare Act

**Constraints** 3
TreatmentBase, seniority (Topic occurrence constraint)
TreatmentBase, indication (Topic occurrence constraint)
TreatmentBase, drug (Topic occurrence constraint)

**Instances**
treatment_Base

Figure 5.7. TreatmentBase in *TM-vMR*

Based on this representation and on the different *MLMs,* medical recommendations were generated like the ones presented in figure 5.8.



Medical Recommendations

The medical advice is: The current treatment is more powerful than the treatment you prescribed. It is advisable to prescribe a more powerful treatment

Patient Data

- Age: 56
- weight: 92
- taille: 178
- sex: M
- HbA1C: 8

Figure 5.8. Treatment recommendation based on *MLMs* translated from *ASTI* rules

## 5.9. Conclusions

In this chapter an analysis of *ASTI* 3 and *Arden Syntax* approaches is presented, in order to increases *CDS* interoperability by mapping medical rules from nonstandard to standard formalisms. Different modalities to translate *ASTI 3* model to *Arden Syntax* were proposed. Each of these methods tries to respond to two major requests: the use of the last version of *Arden Syntax* and usability of obtained *MLMs*. We could see in the first translation that although we respected the entire *Arden Syntax 2.8* version the resulted *MLMs* could not be used in any software which implements this formalism. This translation could only be efficient if version 2.8 of the standard would be fully implemented in software like *Arden2ByteCode*.

The use of multiple *MLMs* is a solution that allows the resulted files (*ASTI 3* to *Arden Syntax* translation) to be executed under various open-source software (*Arden2ByteCode, Egadss*). The test results were presented in the previous section. Another part of the studies regards the link between the data needed for the rules which are represented in *ASTI* (or other formalism in which this rules are translated) and the *vMR* model.

The result of the proposed solutions is the increase of the *CDS* interoperability by allowing the exchange of medical rules between institutions, using *Arden Syntax* standard. Another advantage of translating medical rules which are already in a computer format in standard formalism is the fact that fewer medical specialists are needed for acquiring rules in *Arden Syntax* format, in this way costs would be reduced.

Including the *vMR* as data source for the rules in *ASTI* model can increase the usability of these rules by allowing the connection to any medical information system which implements this standard.

Identified contributions of the author in this chapter are:

- An original analysis of *ASTI* 3 approaches regarding the data model and the computer representation of medical rules in order to develop methods that allow the translation of *ASTI 3* rules in *Arden Syntax* format.
- An analysis of *Arden Syntax* regarding the implementation of complex constructs (e.g., objects), in order to see whether the formalism is complex enough to represent *ASTI* rules.
- Development of three original methods (using *Arden Syntax* objects, multiple *MLMs* and external functions) for the translation of *ASTI 3* rules to *Arden Syntax* in order to respond to: *Arden Syntax* version 2.8, simplicity in understanding, implementation in different software (if the *MLM* can by run in various software)
- Linking *vMR* to *ASTI 3* data model, in order to allow *ASTI 3* to be interconnected with different data sources. A table was built to show which mappings can be done between the two models.
- Adding new elements to the *vMR* data model base on the data needed for the inference in *ASTI 3* software
- validation of *Arden* Rules mapped from *ASTI 3*, for patient datasets regarding diabetes from *Timişoara Emergency County Clinical Hospital,* and comparing the outputs with *ASTI 3* outputs, the recommendations were identical to the ones generated based on *ASTI 3* rules

# 6. CONCLUSIONS

The use of clinical decision support systems has been proven to increase the quality of medical act. Medical rules can be generated, edited, verified, executed (using medical rules in inference engines for reasoning based medical databases) and visualized using computers. Beside the already known advantages of narratives protocols and guidelines (increasing the trust of the medical staff in a medical act or standardization of medical practice) other advantages can be added when implementing computer-based medical protocols and guidelines. These advantages are: faster implementation of new medical knowledge, integration of local databases, decrease of costs, etc. By using clinical decision support systems, large amount of data can be visualized (e.g., blood pressure) easier and intuitively in a graphic manner, for each step of a protocol. Medical protocols and guidelines can be viewed in a graphical manner, avoiding the necessity to read a large amount of data regarding each step of a narrative medical protocol.

To assist medical personnel and patients in decision making, a series of clinical decision support systems have been developed, but all have encountered the same problem: difficulties when integrating in different medical units. To overcome this disadvantage, some of these *CDS* have implemented several of standards for the representation of the input and output data. However, implementation in different institutions remains a major problem [3, 123, 124, 125].

Main problems encountered in the domain of clinical decision support:

- low adaptability of the solution to local databases;
- low interoperability between the clinical decision support systems (different representation formalism);
- several patient profiles and medical situations lack CPGs support.
- 
- In order to address these problems, I suggested several solutions:
- development of an architecture which implements the feature to accept multiple standards (for the representation of patient data) as input, in order to reduce the gap between the representations used in local databases and *CDS*;
- development and implementation of a method for the discovery of medical knowledge and the representation of these rules by using widely accepted standards in order to allow the use of this knowledge in various medical units;
- methods for translating medical rules from a nonstandard formalism to a standard formalism.

These solutions are suggested in Chapters 3-5, accordingly.

In Chapter 2, an analysis of the clinical decision support domain has been made using 44 references. There were presented:

- the reasons for using narrative medical guidelines and protocols;
- the advantages of the computer-based medical guidelines and protocols;
- the main components of a clinical decision support system;

- the most important existing solutions in the domain;
- the evolution of the various approaches;
- a comparison between different approaches;
- the standards used for the representation of patient data;
- the tools used for the visualization of medical recommendations.

In Chapter 3, an architecture with the aim to increase the interoperability of clinical decision support systems by using the *HL 7* standards was developed. The development of the architecture was based on the *Egadss* open-source software. The standards integrated in this architecture were:

- *HL7 CDA*;
- *HL7 vMR*;
- *Arden Syntax*.

The *Data Manager*, *HL7 CDA Component*, *TM-vMR, Egadss* and the *Interface* are the main components of the architecture. They facilitate the communication between the components of the system and medical information systems, allowing the access to multiple data sources types, based on the use of well-known standards.

The patient information contained by these *HL7 CDA* documents and the *TM-vMR* is obtained from different medical sources. When collecting data from various sources using *HL7 CDA* standard and *vMR*, the validity of the patient data is based on the validity of the information stored in the different data bases.

The system was tested for protocols regarding diabetes management, for prescribing the dosage of: insulin, sodium chloride, potassium, glucose. The results of these tests met the physician's requirements.

In Chapter 4, I sugested a new method and a set of associated applications for medical rules discovery and the formalization of these rules in *Arden Syntax* in order to allow their use in various *CDSS*s. This solution aims to increase the *medical rules base,* which is used for the generation of medical advices. The technologies used are:

- *data mining* for the discovery of medical rules;
- *Arden Syntax* for the representation of these rules.

A solution for translating the medical rules obtained from databases using *data mining* technology (with the open-source software *WEKA*) to rules represented in *Arden Syntax* language (obtaining new *MLMs*) was developed. These new rules obtained from the databases are not evidence-based. In this respect, each new *MLM* should be validated by medical experts. The next step in improving the acceptance of the new recommendation is testing this recommendation in medical units.

Another aspect of this work regards the fact that until now the translation was tested only for rules regarding the newborns databases and just 3 types of rules were obtained. In this research, we suggest a method for the translation of *data mining* rules to *Arden Syntax*. The main reason for using *Arden Syntax* for the representation of medical rules is that it is the most used standard in the domain of *CDS* [16, 48, 49].

In Chapter 5, a comparison between different formalization approaches for the representation of medical rules (*ASTI* and *Arden Syntax*) is presented. This comparison resulted in several methods for translating medical rules from a formalism (not accepted as a standard) to another (accepted as a standard). The translation of medical rules is important in order to acquire new knowledge in a standard format and to allow the interchanges of this knowledge between institutions.

The new rules were tasted for the use in conjunction with the *vMR* data model. This study showed that the *vMR* does not allow the representation of all medical data needed for the generation of recommendations. In order to allow the representation of such information, I suggested new components, as an extension to be integrated in the *vMR* data model.

The use of *HL7* standards eases the integration of the system in multiple medical units. In this way, one of the major problems, the local adaptation of clinical decision support systems can be resolved. Also, the acquisition of medical rules in *Arden Syntax* format (by translating them from *data mining* rules or from other formalisms) increases the interoperability between clinical decision support systems.

***Contributions***

By achieving the proposed objectives several contributions are claimed by the author. These contributions are classified and presented in the next paragraphs.

**A. Contributions** to the **analysis** and presentation of various aspects of the field:

- An original assessment of the current level of knowledge and development achieved in the field of clinical decision support, underlining its importance in improving the medical care act.
- The assessment of several of the most important existing approaches related to the generation of medical recommendations in terms of the two main modules of clinical decision support systems (authoring tools and execution engine).
- An original analysis of *ASTI 3* approach regarding the data model and the computer representation of medical rules;
- An analysis of *Arden Syntax* regarding the implementation of different constructs (e.g., objects) in various software solutions;
- Analysis of the *HIPAA* regulation in other to implement a secure communication between the different components of the system with consequences in future enabling secure communication in my application;
- The assessment of the evolution of different *CDS* approaches in computer-based guidelines and protocols, starting from 1975 to 2012, with my own the improvement of an existing time-oriented graphical representation of the history of evolution of *CDSSs.*

B. **Theoretical** and **methodological** contributions:

- Development of a method which allows translation of *data mining* rules into specific *Arden Syntax* files (*MLMs*) in order to make the knowledge discovered with the help of the *data mining technology* more accessible to clinical decision support systems (through the representation of knowledge with the help of a standard);

- Categorization of the elements of the rules in: *variable, value, operator, conclusion_variable, conclusion_value*. This categorization of the variables was necessary in order to generate the different *MLM's* sections in the knowledge discovery process;
- Development of three different methods (using *Arden Syntax* objects, multiple *MLMs* and external functions) that permit the translation of *ASTI* specific constructions (e.g. medication objects representing) in *Arden Syntax*;
- Mapping *vMR* to *ASTI 3* data model, in order to allow *ASTI* to be interconnected with different data sources;
- Adding new elements to the *vMR* data model based on the patient data needed for the inference in *ASTI 3* software.

C. Contributions to the development of original **software solutions**:

- Proposal, development, implementation and validation of a system architecture that allows the integration of multiple data sources based on *HL7* standards, by extending *Egadss* software by adding a new module - *Data Manager* - designed to implement these communication features;
- Development of an innovative software tool to connect the *CDSS* to the *TM-vMR* database;
- Development of an interface to implement certain features well appreciated by physicians (specific clinical decision support systems), based on a previous analysis of the existing solutions;
- Development of all modules belonging to the architecture used for knowledge discovery through *data mining* technology and *Arden Syntax*.

D. Contributions to **validation** and **evaluation** of proposed solutions based on real life case studies:

- The evaluation of the interfaces of existing computer-based guideline solutions by using an original questionnaire-based approach and the development of a method for the evaluation of the user interface quality;
- Development of a methodology for validation of the use of multiple *HL7 CDA* documents and *vMR* and applying this in a real-life case study with 30 patient datasets for the management of diabetes used in *Timişoara Emergency County Clinical Hospital;*
- Development of a methodology for validation of new medical rules (in *Arden Syntax*) obtained using *Data mining,* and applying this in a real-life case study with datasets for newborns*, from Obstetrics and Gynecology department of Bega clinic*, *Timișoara;*
- Development of a methodology for validation of rules in *Arden Syntax* format mapped from *ASTI 3*, and applying this in a real-life case study with with patient datasets regarding diabetes from *Timişoara Emergency County Clinical Hospital,* and comparing the generated recommendations with *ASTI 3* outputs*.*

I hope these contributions lead to the development of *CDSSs* which have access to complex medical information and generated more complete and precise recommendation in order to improve the medical act.

***Future developments***

Regarding the research presented in Chapter 3, the future implementation of other standards for patient data representation in the proposed architecture could increase the adaptability of the system to local context. This could be done by adding other modules (in the *Data Manager*) which should deal, for example, with data in *CCD* (Continuity of Care Document) format.

In order to extend the capabilities of the system architecture presented in Chapter 4, other methods for knowledge discovery could be implemented, for extracting medical rules. The author will use the developed technology for other databases than the one used in this research, in order to obtain more specific and needed recommendations to complete the gaps from different guidelines. A further development would be the implementation of validation methods (involving medical specialists) that would permit the new medical rules to be accepted as evidence-based.

Regarding the translation methods of medical rules proposed in Chapter 5, other formats for representation of medical rules should be analyzed in order to develop new software tools which permit the translation of these formats to formalisms accepted as standards (*Gello* and *Arden Syntax*). The result would improve the communication between clinical decision support systems. It would also be important to analyze the data needed for different approaches (which were omitted in the development of the *HL7 vMR*) and if they are available in the *vMR* data model (which is the only standard for the representation of patient data for clinical decision support systems). This analysis might lead to other data elements which need to be added to the *vMR* model. Another research direction could be the development of services which enable the communication between *ASTI 3* software and medical information systems that implement *vMR* standard, in order to allow *ASTI 3* to communicate with multiple data sources.

The implementation of the suggested solutions in medical units help the physicians to increase the quality of medical care, give more efficient treatments and use new medical knowledge in their current clinical practice.

# REFERENCES

[1] Clercq, P., Kaiser, K.,Hasman, A., ”Computer-Interpretable Guideline Formalisms”, Computer-based Medical Guidelines and Protocols: A Primer and Current Trend, A. ten Teije, S. Miksch and P.J. Lucas, vol. 139, pp. 22-43, 2008.

[2] Rosenbrand, K., Croonenborg, J., Wittenberg,J., ”Guideline Development”, Computer-based Medical Guidelines and Protocols: A Primer and Current Trend, A. ten Teije, S. Miksch and P.J. Lucas, vol. 139, pp. 3 -21, 2008.

[3] Groot, P.,Hommersom, A.,Lucas, P., “Adaptation of Clinical Practice Guidelines”, Computer-based Medical Guidelines and Protocols: A Primer and Current Trend, A. ten Teije, S. Miksch and P.J. Lucas, vol. 139, 2008, pp. 121-139.

[4] Shiffman R. N., Liaw Y., Brandt C.A., Corb G.J., “Computer-based guideline implementation systems: a systematic review of functionality and effectiveness” J Am Med Inform Assoc. Vol. 6, pp. 104-114, 1999.

[5] Chesani, F., Lamma, E., Mello, P., Montali, M., Storari, S., Baldazzi P., Manfredi, M., “Compliance Checking of Cancer-Screening CareFlows: An Approach Based on Computational Logic”, Computer-based Medical Guidelines and Protocols: A Primer and Current Trends, vol. 139, pp. 183-192, 2008.

[6] The Healthcare Battle for a Clinical Decision Support System (CDSS) Control Point Begins http://www.scientiaadv.com/blog/2010/12/16/the-healthcare-battle-for-a-clinical-decision-support-system-CDSS-control-point-begins/, accessed in 23.06.2012.

[7] Latoszek-Berendsen, A., Tange, H., Herik, H.J., Hasman,A., “From clinical practice guidelines to computer-interpretable guidelines. A literature overview.” in Methods of Information in Medicine (2010), vol. 49, Issue 6, pp. 550-570, 2010.

[8] Kawamoto K., Houlihan C.A., Balas E.A., Lobach D.F., “Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success” in BMJ, pp. 765-768, 2005.

[9] Hunt D.L., Haynes R.B., Hanna S.E., Smith K., “Effects of computer-based clinical decision support systems on physician behaviour and patient outcomes: A systematic review” JAMA, Vol. pp. 1339-1346, 1998.

[10] CDA – Clinical Document Architecture, http://www.corepointhealth.com/resource-center/HL7- resources/HL7-CDA, accesed in 6.06.2010.

[11] Kawamoto, K., Virtual Medical Record (vMR) for Clinical Decision Support– Domain Analysis Model – HL7 Project #184 Informative Ballot September 2011 - model propus spre standardizare, 2011.

[12] http://web.cs.dal.ca/~sraza/StudentWork/Gaston%20paper.PDF - Li, M., Superceanu, B., Zheng, Z. ; "GASTON: A GENERIC FRAMEWORK FOR CLINICAL GUIDELINE DEVELOPMENT" – accessed in 03.03.2011.

[13] Peek, N., Goud, R., de Keizer, N., van Engen-Verheul, M., Kemps, H., Hasman, A.; CARDSS: Development and evaluation of a guideline based decision support system for cardiac rehabilitation, Artificial Intelligence in Medicine (AIME) Conference, Bled, Slovenia, July 2011.

[14] SAGE, Computer-based protocols and guidelines, the Sage approach, http://www.openclinical.org/gmm_sage.html - accessed in 02.09.2010.

[15] Simon, A.C.R., Holleman, F., Hoekstra, J.B. , De Clercq, P.A., Lemkes, B.A., Hermanides, J., Peek, N.; "Development of a web-based decision support system for insulin self-titration", Medical Informatics Europe (MIE) Conference 2011, Oslo, Norway, August 2011.

[16] Weber-Jahnke, J. H., McCallum, G., A light-weight component for adding decision support to electronic medical records, Hawaii International Conference on System Sciences, Proceedings of the 41st Annual, pp. 251 – 251, ISBN: 978-0-7695-3075-8, 2008.

[17] Peleg, M., Tu, S., Bury, J., Ciccarese, P., Fox, J., Greenes, R., Hall, R., Johnson, P., Jones, N., Kumar, A., Miksch, S., Quaglini, S., Seyfang, A., Shortliffe, E., Stefanelli, M., "Comparing computer interpretable guideline models: a case-study approach" J Am Med Inform Assoc, pp. 52-68, 2003.

[18] Peleg, M., Wang, D., Fodor, A., Keren, S., Karnieli, E., "Lessons Learned from Adapting a Generic Narrative Diabetic-Foot Guideline to an Institutional Decision-Support System", Computer-based Medical Guidelines and Protocols: A Primer and Current Trends, vol. 139, pp. 243-252, 2008.

[19] Clercq, P. A., Blomb, J. A., Hasman, A., Korstenb, H. M., "GASTON: an architecture for the acquisition and execution of clinical guideline-application tasks" Med Inform Internet Med, pp. 247-63, 2000.

[20] Tan, J., "Adaptive Health Management Information Systems – Concepts, Cases and Practical Applications", Jones and Bartlett Publishers, pp 156-161, 2010.

[21] Bilykh, I., Jahnke, J. H., McCallum, G., Price, M., Using the Clinical Document Architecture as open data exchange format for interfacing EMRs with clinical decision support systems, Proceedings of the 19th Symposium on Computer-Based Medical Systems (CBMS'06), ISBN: 0-7695-2517-1, 2006.

[22] Computer-based guidelines and protocols, http//:www.openclinical.org, accessed in 10.03.2010.

[23] Arden Syntax- http://www.HL7.org/documentcenter/public_temp_D666E87D-1C23-BA17-0CD9B405AFC9B718/calendarofevents/himss/2011/HL7%20Organizational%20Backgrounder%20and%20Standards%20Descriptions.pdf – 04.02.2010

[24] Şerban, P., Mircescu, Ş., Papuc, R., Poteraş, A. M., Nicolae, S., Câmpean, G., Şoavă, M., Mariuţan, S. G., "Durerea lombară nespecificată a adultului", Ghid de pactică pentru medici de familie, Editura Infomedica, ISBN 973-7912-44-6, 2005.

[25] Seroussi B., Bouaud J., Denke D.L., Falcoff H., Julien J., "Using knowledge modeling to measure how clinical practice could actually be evidence-based: apreliminary analysis with arterial hypertension management", Stud Health Technol inform., Vol. 150, pp. 668-672, 2009.

[26] Gortzis, L.G., Nikiforidis, G., "Tracing and cataloguing knowledge in an e-health cardiology environment", Journal of Biomedical Informatics, Volume 41, Issue 2, pp. 217-223, April 2008.

[27] ANSI/HL7 Arden V2.8-2008, approved in March 13, 2012, The Health Level Seven Arden Syntax Version 2.8.

[28] Virtual Medical Record (vMR) for Clinical Decision Support – Domain Analysis Model, http://wiki.HL7.org/images/archive/6/6b/20110729073300!HL7vMR_vMR_Domain_Analysis_Model_2011_Sept_Ballot.pdf, accessed in 05.02.2012.

[29] Simonaitis L., Belsito A., Cravens G., Shen C., Overhage J.M., "Continuity of Care Document (CCD) Enables Delivery of Medication Histories to the Primary Care Clinician", AMIA Annual Symposium Proceedings, pp. 747-751, 2010.

[30] Fox, J., Black, E., Chronakis, I.,Dunlop, R.,Patkar, V.,South, M.,Thomson, R., "From Guidelines to Careflows: Modelling and Supporting Complex Clinical Processes", Computer-based Medical Guidelines and Protocols: A Primer and Current Trend, A. ten Teije, S. Miksch and P.J. Lucas, vol. 139pp. 44-62, 2008.

[31] Wang, D., Pelegb, M., Bu, D.,Cantor, M.,Landesberg, G.,Lunenfeld, E.,Tu, S. W.,Kaiser, G. E.,Hripcsak, G.,Patel, V. L.,Shortliffe, E. H.,"GESDOR – A Generic Execution Model for Sharing of Computer-Interpretable Clinical Practice Guidelines", AMIA Annu Symp Proc. 2003; pp. 694–698

[32] Wang, D., MorPeleg, Tub, S. W., Boxwalac, A. A., Ogunyemic, O., Zengc, Q., Greenesc, R. A., Patela V. L. and Shortliffea E. H., Design and implementation of the GLIF3 guideline execution engine. Journal of biomedical Informatics, Volume 37, Issue 5, Pages 305-318, October 2004.

[33] Bernad, E., „Data sistematization and management in obstetrics and gynecology" (Sistematizarea şi managementul datelor în obstetric şi ginecologie), ed. Artpress, 2009, ISBN 978-973-108-230-1, 2009.

[34] Quaglini, S.,"Compliance with Clinical Practice Guidelines", Computer-based Medical Guidelines and Protocols: A Primer and Current Trend, A. ten Teije, S. Miksch and P.J. Lucas, vol. 139, pp. 160-179, 2008.

[35] Hommersom, A., Groot, P., Lucas, P., Marcos, M., Martínez-Salvador, B., "A Constraint-Based Approach to Medical Guidelines and Protocols", Computer-based Medical Guidelines and Protocols: A Primer and Current Trends, vol. 139, pp. 213-222, 2008.

[36] Schmitt, J., Balser, M., Reif, W.,"Verification of Medical Guidelines in KIV", Computer-based Medical Guidelines and Protocols: A Primer and Current Trends, vol. 139, pp. 253-263, 2008.

[37] HL7 Canada, http://sl.infoway-inforoute.ca, accessed in 05.06.2010.

[38] HL7 version 3 Standard: Abstract Transport Specification, www.HL7.org/v3ballot/html/downloads/downloads.htm, 2010.

[39] Kawamoto K., "Integration of knowledge resources into applications to enable clinical decision support: architectural considerations." Greenes RA, Clinical Decision Support: the Road Ahead. Boston: Elsevier, pp. 503-538, 2007.

[40] Osheroff J.A. et al, "A roadmap for national action on clinical decision support", in J Am Med Inform Assoc., pp. 141-145, 2007

[41] K. Kawamoto, Multi-National, Multi-Institutional Analysis of Clinical Decision Support Data Needs to Inform Development of the HL7 Virtual Medical Record Standard; AMIA Annu Symp Proc.; pp. 377–381, 2010.

[42] Kawamoto, K., Virtual Medical Record (vMR) for Clinical Decision Support – GELLO Implementation Guide -, HL7 Project #18420 Draft Standard for Trial Use 21 September 2011.

[43] Cerny R., "Topincs: A software for rapid development of web databases", http://www.cerny-online.com/documents/Topincs%20-%20A%20software%20for%20rapid%20development%20of%20web%20databases.pdf, accessed in 21.12.2011.

[44] Grieve G., et al, HL7 Version 3 Standard: Abstract Transport Specification, www.HL7.org, Accessed in 01.06.2012.

[45] Healthcare Information and Management Systems Society Electronic Health Record Vendor Association (EHRVA), Quick Start Guide, HL7 Implementation Guide: CDA Release 2 – Continuity of Care Document (CCD), 2007.

[46] Clinical Decision Support & Arden Syntax Technical Committee of HL7, inventor Arden Syntax for Medical Logic Systems, version 2.0. Draft revision. USA. July 7, 1999.

[47] ANSI/HL7 Arden V2.7-2008, December 10, 2008 The Health Level seven Arden Syntax Version 2.7.

[48] Gietzelt, M. et all, "ARDEN2BYTECODE: a one-pass Arden Syntax compiler for service-oriented decision support systems based on the OSGi platform", Comput Methods Programs Biomed journal , vol. 106, ISSN 0169-2607, pp. 114-125, 2012.

[49] Fehre K., Adlassnig K.-P., „Service-Oriented Arden-Syntax-Based Clinical Decision Support" Schreier, G., Hayn, D., and Ammenwerth, E. (Eds.) Proceedings of the eHealth2011, Austrian Computer Society (Tagungsband der eHealth2011, Österreichische Computer Gesellschaft), Wien, 123–128, 2011.

[50] Sordo M., Ogunyemi O., Boxwala A. A., Greenes R. A., "GELLO: An Object-Oriented Query and Expression Language for Clinical Decision Support", AMIA Annu Symp Proc. 2003, pp.1012, 2003.

[51] Sordo M, Boxwala A. A., Ogunyemi O, Greenes R.A., "Description and status update on GELLO: a proposed standardized object-oriented expression language for clinical decision support", Stud Health Technol Inform. 2004, vol. 107, pp. 164-168, 2004.

[52] Normativ HL7 standard - Gello.

[53] Hatsek, A., Young, O., Shalom, E., Shahar, Y., "DeGeL: A Clinical-Guidelines Library and Automated Guideline-Support Tools", Computer-based Medical Guidelines and Protocols: A Primer and Current Trends, vol. 139, 2008, pp. 203-212

[54] Computer-based protocols and guidelines. GLIF, http://www.openclinical.org/gmm_GLIF.html, 01.12.2009.

[55] Asbru, http://www.asgaard.tuwien.ac.at/plan_representation/Asbru_doc.html, accessed in 01.03.2010.

[56] Gmm, Computer-based protocols and guidelines, Asbru, http://www.openclinical.org/gmm_Asbru.html, 22.11.2009.

[57] Computer-based protocols and guidelines, PROforma, http://www.cossac.org/technologies/PROforma, 15.11.2009.

[58] Lamy J. et al., "A generic system for critiquing physicians'prescriptions: usability, satisfaction and lessons learnt", Studies in Health Technology and Informatics, vol. 169, pp. 125-129, 2011.

[59] Hunter, J., "TSNet – A Distributed Architecture for Time Series Analysis", Computer-based Medical Guidelines and Protocols: A Primer and Current Trends, vol. 139,, pp. 223-232, 2008.

[60] Clercq, P. A., Blomb, J. A., Hasman, A., Korstenb, H. M., "Design and implementation of a framework to support the development of clinical guidelines" Int J Med Inf, pp. 285-318, 2001.

[61] Hommersom, A., Groot, P., Balser, M., Lucas P., "Formal Methods for Verification of Clinical Practice Guidelines", Computer-based Medical Guidelines and Protocols: A Primer and Current Trends, vol. 139, pp. 63-80, 2008.

[62] Terenziani, P., German, E., Shahar, Y., "The Temporal Aspects of Clinical Guidelines", Computer-based Medical Guidelines and Protocols: A Primer and Current Trends, vol. 139, pp. 81-100, 2008.

[63] Anselma, L., Montani, S., "Planning: Supporting and Optimizing Clinical Guidelines Execution", Computer-based Medical Guidelines and Protocols: A Primer and Current Trends, vol. 139, pp. 100-120, 2008.

[64] Domínguez, D., Fernández, C., Meneu, T., Mocholí, J., Serafin, R., "Medical Guidelines for the atient: Introducing the Life Assistance Protocols", Computer-based Medical Guidelines and Protocols: A Primer and Current Trends, vol. 139, pp. 93-202, 2008.

[65] Seyfang, A., Paesold, M., Votruba, P., Miksch, P,"Improving the Execution of Clinical Guidelines and Temporal Data Abstraction in High-Frequency Domains", Computer-based Medical Guidelines and Protocols: A Primer and Current Trends, vol. 139, pp. 100-120, 2008.

[66] Leong, T.-Y., Kaiser, K., Miksch, S., "Free and Open-source Enabling Technologies for Patient-Centric, Guideline-Based Clinical Decision Support: A Survey", Yearb Med Inform, pp. 74–86, 2007.

[67] Aigner, W., Miksch, S., "CareVis: Integrated visualization of computerized protocols and temporal patient data" Artificial Intelligence in Medicine, Volume 37, Issue 3, pp. 203-218, July 2006.

[68] Kosara, R., Miksch, S., Seyfang, A., Votruba, P., "Tools for Acquiring Clinical Guidelines in Asbru", June, 2002.

[69] Clercq, P. A., Blomb, J. A., Korstenb, H. M., Hasman, A., "Approaches for creating computer-interpretable guidelines that facilitate decision support", Artificial Intelligence in Medicine, Volume 31, Issue 1, pp. 1-27, May 2004.

[70] Fox, J., Johns, N., Rahmanzadeh, A., "Disseminating medical knowledge: the PROforma approach", Artificial Intelligence in Medicine,Volume 14, Issues 1-2, pp. 157-182, September-October 1998.

[71] Glee, http://people.dBMI.columbia.edu/homepages/wandong/homepage20030326_files/GLEE.htm, 01.04.2010.

[72] Lamy J. et al., "Use of the C4.5 machine learning algorithm to test a clinical guideline-based decision support system", in Proc. MIE, pp.223-228, 2008.

[73] Lamy J. et al., "How to translate therapeutic recommendations in clinical practice guidelines into rules for critiquing physician prescriptions? Methods and

application to five guidelines", BMC Medical Informatics and Decision Making journal, vol. 10, pp. 31-44, 2010.

[74] Aigner, W., Kaiser, K., Miksch, S., "Visualization Methods to Support Guideline-Based Care Management", Computer-based Medical Guidelines and Protocols: A Primer and Current Trends, vol. 139, pp. 140-159, 2008.

[75] Asbru View, http://www.asgaard.tuwien.ac.at/tools/asbruview.html, accessed in 20.07.2012.

[76] Gomoi, V.,  Stoicu-Tivadar, V., "Evaluation of computer-based medical protocols and guidelines approaches", 8th ICICTH Samos, Greece, ISBN: 978-960-466-095-9, pp. 151-160, 2010.

[77] Wiley, J.,"Interaction Design: beyond human-computer interaction", pp. 585-684. ISBN 978-0-470-01866-8, 2007.

[78] Li, Z., Alaeddine, N., Tian, J., "Multi-faceted quality and defect measurement for web software  and source contents", Journal of Systems and Software, Volume 83, Issue 1, pp. 18-28, January 2010.

[79] Vida, M., Gomoi, V., Stoicu – Tivadar, L., Stoicu-Tivadar, V., "Generating medical computer-based protocols using standardized data transmission", Soft Computing Applications (SOFA), 2010 4th International Workshop, pp. 155 – 158 2010.

[80] Vida, M., Gomoi V., Stoicu-Tivadar L., Stoicu-Tivadar V., "Using HL7 CDA Standard to Connect Medical Databases as a Support for Automatic Generation of Computer Interpretable Guidelines", Proceedings of the 31st National Conference on Medical Informatics, pp. 69-74, 2010.

[81] Gomoi, V.,  Stoicu-Tivadar, V., "A new visualization solution for medical computer-based protocols", 9th ICICTH, 2011.

[82] V. Gomoi, M. Vida. L. Stoicu-Tivadar, V. Stoicu-Tivadar, "Extracting clinical information to support medical decision based on standards", in EFMI STC 2011 Conference, Lasko, Slovenia, 14-15 April 2010.

[83] International Classification of Disease (ICD), http://www.who.int, Accessed in 11.10.2010.

[84] McCallum. G., "EGADSS: A Clinical Decision Support System for use in a Service-oriented Architecture", MSc. Thesis, Computer Science, University of Victoria, BC, Canada, 2006.

[85] Egadss, http://www.Egadss.org/, 15.08.2010.

[86] International Classification of Disease (ICD), http://www.who.int, Accessed in 17.09.2011.

[87] Turrentine, J. E., Clinical protocols in obstetrics and gynecology, third edition, ISBN-13: 978 0 415 43996 1, pp. 82-85, 2008.

[88] Metro technology, http://www.ibm.com/developerworks/java/library/j-jws9/index.html - accessed in 21.01.2012.

[89] Biezunski, M,. Bryan, Newcomb, S., "Topic Maps ISO/IEC 13250-2:2006", 2006.

[90] Librelotto, G. R. , Ramalho, J. C. , Henriques, P. R. , "XML Topic Map Builder: Specification and Generation" in XATA: XML, Aplicações e Tecnologias Associadasactas, 1, Braga, 2003.

[91] Pepper, S., "Topic Maps", Department of Linguistics, University of Oslo, Norway http://www.ontopedia.Net/pepper/papers/ELIS-TopicMaps.pdf, accesed in 20.01.2011.

[92] Gruber, T., "Ontology", Encyclopedia of Database Systems, Ling L. and M. T. Özsu (Eds.), Springer-Verlag, 2009.

[93] Topic map query language, http://www.isotopicmaps.org/tmql/, accessed in 10.12.2010.

[94] Topic map - http://XML.coverpages.org/TMQL-18048-FCD-2008-07-15.pdf - accessed in 01.02.2011.

[95] Dragu D., Gomoi V., Stoicu-Tivadar V., "Automatic generation of medical recommendations using Topic Maps as knowledge source", 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI), SACI2011, Timişoara, Romania, ISBN: 978-1-4244-9108-7, pp. 593 – 597, 2011.

[96] NuSOAP technology, http://www.scottnichol.com/NuSOAP intro.htm, accessed in 18.02.2012.

[97] Patkar, V., Fox, J., "Clinical Guidelines and Care Pathways: A Case Study Applying PROforma Decision Support Technology to the Breast Cancer Care Pathway", Computer-based Medical Guidelines and Protocols: A Primer and Current Trends, vol. 139, pp. 233-242, 2008.

[98] Terenziani, P., Montani, S., Bottrighi, A., Molino, G., Torchio, M., "Applying Artificial Intelligence to Clinical Guidelines: The GLARE Approach", Computer-based Medical Guidelines and Protocols: A Primer and Current Trends, vol. 139, pp. 121-139, 2008.

[99] HIPAA, U.S. Department of Health & Human Services, http://www.hhs.gov, Accessed in 10.10.2011.

[100] OCR HIPAA Privacy, Disclosures for public health activities [45 CFR 164.512(b)], 2002.

[101] BlueCross BlueShield of Tennessee, HIPAA Portability & Accountability: How HIPPA Affects Individual Coverage, http://www.bcbst.com/, Accessed in 12.10.2011.

[102] The Advantages of HIPAA, http://www.ehow.com, Accessed in 12.10.2011.

[103] Shen J. J., Samson L. F., Washington E. L., Johnson P., "Barriers of HIPAA Regulation to Implementation of Health Services Research", Journal of Medical Systems, Vol. 30, pp. 65-69, 2006.

[104] Choi B. Y., Capitan K. E., Krause J. S., Streeper M. M., "Challenges Associated with Privacy in Health Care Industry: Implementation of HIPAA and the Security Rules", Journal of Medical Systems, Volume 30, pp. 65-69, 2006.

[105] HIPAA disadvantages, http://www.finweb.com/insurance/5-disadvantages-of-HIPAA.html, accessed in 10.10.2011.

[106] HIPAA disadvantages, http://www.ehow.com/list_7370337_HIPAA-disadvantages.html, accessed in 10.10.2011.

[107] Lee C.-D., Ho K.I., Lee W.-B., "A Novel Key Management Solution for Reinforcing Compliance With HIPAA Privacy/Security Regulation" in Information Technology in Biomedicine, IEEE Transactions , 2011 Jul; ISSN: 1089-7771, pp. 550-556, 2011.

[108] Li J., Lee J.-S., Chang C.-C., "Preserving PHI in Compliance with HIPAA Privacy/Security Regulations Using Cryptographic Techniques", IIHMSP '08 International Conference, ISBN: 978-0-7695-3278-3, pp. 1545 – 1548, 2008.

[109] Hardin M.J., Chhieng D.C., „Data mining and Clinical Decision Support Systems.", Berner S, editor. Clinical Decision Support Systems Theory and Practice. New York: Springer; pp. 44-63. 2007.

[110] Bennett C., et al. „Data mining Session-Based Patient Reported Outcomes (PROs) in a Mental Health Setting: Toward Data-Driven Clinical Decision Support and Personalized Treatment", Proceeding of the IEEE First International Conference on Healthcare Informatics, Imaging and Systems Biology; 2011 Jul 26-29; San Jose, USA. Washington: IEEE Computer Society, pp. 229-236, 2011.

[111] Zhuang Z.Y., Churilov L., Burstein F., Sikaris K., „Combining data mining and case-based reasoning for intelligent decision support for pathology ordering by general practitioners", European Journal of Operational Research. Vol 195, pp. 662-675, 2009.

[112] Gomoi, V., Stoicu-Tivadar, V., "A new method in automatic generation of medical protocols using artificial intelligence tools and a Data Manager", ICCC-CONTI, 2010.

[113] Gomoi V., Robul R., Stoicu-Tivadar V., Bernard E., "From data mining rules to Medical Logical Module and medical recommendations", in press.

[114] Parpinelli R.S., Lopes H.S., Freitas A.A., „Data mining with an ant colony optimization algorithm", IEEE Transactions on Evolutionary Computation., Vol 6 pp. 321-32, 2010.

[115] Robu R., Stoicu-Tivadar V., „Arff Convertor Tool for WEKA Data mining Software", Proceeding of the IEEE International Joint Conferences on Computational Cybernetics and Technical Informatics; pp. 247-251, 2010.

[116] Hall M., et al, „The WEKA Data mining Software: An Update" ACM SIGKDD Explorations Newsletter, Vol 11, pp. 10-18, 2009.

[117] Frank, E. et al, "WEKA", Data mining and Knowledge Discovery Handbook, Springer, Maimon, O., Rokach L., pp. 1305-1314, 2005.

[118] HL7 Clinical Document Architecture, Release 2.0, HL7 version 3 Interoperability Standards, Normative Edition 2009, Disk 1 – Standards Publication, 2009.

[119] CDA levels, http://www.corepointhealth.com, Accessed in 05.06.2010.

[120] Chiu R., Chang K., Tsai K., "An implementation for Healthcare Information Delivery System in Adopting HL7 v3 and CDA standards and Cutting – Edge Information Technologies", www.healthlab.im.fju.edu.tw/Publications, 2006, Accessed in 06.04.2010.

[121] Cornet R., de Keizer N., "Forty years of SNOMED: a literature review", BMC Med Inform Decis Mak journal, vol. 10, pp. S2, 2008.

[122] Windows communication foundation - http://msdn.microsoft.com/en-us/library/ms731082.aspx, accesed in 20.03.2012

[123] Penney, G., Foy, R., „Do clinical guidelines enhance safe practice in obstetrics and gynecology", Best Practice & Research Clinical Obstetrics and Gynecology, Vol. 21, No. 4, pp. 657-673, 2007.

[124] GLIF: a Language for Sharing and Executing Clinical Guidelines Mor Peleg, Ph.D. Post-doctoral Fellow, Stanford University, CA, http://www.openclinical.org/docs/ext/workshops/w1/pelegGLIF.pdf, 01.05.2010.

[125] Ahmadiana, L., van Engen-Verheula, M., Bakhshi-Raieza, F., Peek, N., Corneta, R., de Keizer, N. F.,; "The role of standardized data and terminological systems in computerized clinical decision support systems: Literature review and survey", International Journal of Medical Informatics, Special Issue: Security in Health Information Systems, Volume 80, Issue 2, pp. 81-93, February 2011.