Design of a Goal Ontology for Medical Decision-Support

Davide Zacacagnini

M.D

Submitted to the Department of Health Sciences and Technology
in Partial Fulfillment of the Requirement for the Degree of
Masters of Science in Biomedical Informatics

at the
Massachusetts Institute of Technology

August 2005
[September 2005]

Signature of Author: [illegible]
Department of Health Sciences and Technology
August 14, 2005

Certified By: [illegible]
Aziz Boxwala
Clinical Assistant Professor of Radiology, HMS
Affiliated Faculty, Division of Health Sciences and Technology
Thesis Supervisor

Accepted By: [illegible]
Martha L. Gray
Director, HST
Edward Hood Taplin Professor of Medical and Electrical Engineering, MIT

-blank page-

ABSTRACT

Objectives: There are several ongoing efforts aimed at developing formal models of medical knowledge and reasoning to design decision-support systems. Until now, these efforts have focused primarily on representing content of clinical guidelines and their logical structure. The present study aims to develop a computable representation of health-care providers' intentions to be used as part of a framework for implementing clinical decision-support systems. Our goal is to create an ontology that supports retrieval of plans based on the intentions or goals of the clinician.

Methods: We developed an ontological representation of medical goals, plans, clinical scenarios and other relevant entities in medical decision-making. We used the resulting ontology along with an external ontology inference engine to simulate selection of clinical recommendations based on goals. The ontology instances used in the simulation were modeled from two clinical guidelines.

Testing the design: Thirty-two clinical recommendations were encoded in the experimental model. Nine test cases were created to verify the ability of the model to retrieve the plans. For all nine cases, plans were successfully retrieved.

Conclusion: The ontological design we developed supported effective reasoning over a medical knowledge base. The immediate extension of this approach to be fully developed in medical applications may be partially limited by the lack of available editing tools. Many efforts in this area are currently aiming to the development of needed technologies.

Thesis Supervisor: Aziz Boxwala

Title: Clinical Assistant Professor of Radiology, HMS

Acknowledgments

This text is the result of a long journey started far away, about three years ago. Many people accompanied me along the way, some of them physically, others from the distance with their minds and love. To them goes my absolute gratitude.

Aziz Boxwala deserves my highest recognition for having been a supportive and enlightening advisor.

TABLE OF CONTENTS

Chapter 1. Background

Section 1.1. Introduction

The aim of the present study was to develop an ontological representation of healthcare providers' intentions, or goals that can be used in the implementation of decision-support systems in medicine. We built an ontology that provides a formal definition of clinicians' aims in the process of caring for patients. We built also a knowledge base of care plans containing specified goals. Using a set of test patient cases, we verified our system's capability to select relevant and useful plans of action according to goals and patients' clinical state.

The importance of formal definition of goals has already been recognized in other fields, such as robotics, to enable an agent to choose consistently plans of actions appropriate to the scenario. We believe that a system with such capability would be instrumental to automate different tasks such as information retrieval, task scheduling and, more generally, decision-support when embedded in a clinical information system. Limited work has been done in the medical domain to investigate such a capability, i.e. intentional planning, using the ontological approach.

We used the Protégé-OWL environment that supports ontology editing and description logic based reasoning to implement our system. In the current version, plans have been chosen from two clinical guidelines and modeled using the Hierarchical Task Decomposition approach, a well established technique to build plans libraries. These guidelines were for the diagnosis of breast disease and hypertension diagnosis and treatment. Both the guidelines were published by the Institute for Clinical Systems Improvement (ICSI).

The possibility of creating highly scalable knowledge bases, that is, plan libraries, containing multiple and heterogeneous clinical action plans, while easily allowing consistency checking and robustness, is one of the more attractive feature of this approach. Our design has been demonstrated to be effective in executing complex reasoning over the knowledge base and selecting relevant medical plans consistent with clinical scenario and stated providers' intentions.

Section 1.2. Ontology

1.2.1 Ontology and its applications in software development

Software engineering, as many pointed out, can be considered fundamentally a knowledge representation activity[1] [2]. Ontological representations are capable of formally defining and classifying objects, being them tangible ob abstract, existing in a portion of reality and the relations among them.

Interest in ontology development and application in computer science have recently gained energy because of current trends in software development. Companies and public institutions increasingly rely on their computerized information systems to deliver services and products. Old software architectures, with limited scopes and capabilities, are therefore forced to evolve into articulated systems capable of tracking abstract or physical objects along different processes, integrating different users' views and allowing their interaction, while setting clear boundaries to ensure security and efficiency and bringing together disparate pieces of information. In other words, the complexity of information systems is increasing[3]. The current challenge is a peculiar one. An increasing amount of knowledge gathered from disparate areas needs in fact to be incorporated in this new generation of applications in order to answer public demands[4]. Ontology plays an important role in transferring knowledge from human to computers and therefore allowing this new generation of software to be born.

1.2.2 The Semantic Web

Furthermore Ontology is also been chosen by the World Wide Web Consortium (W3C) as the way of representing semantic information in the future world wide web, i.e. the Semantic Web[5].

This new architecture will add a range of novel capabilities to the Internet, which are still to be fully envisioned. In the Semantic Web, information will be encoded as ontologies in web pages that will be read and interpreted by programs. These programs in turn will be able to perform much of what currently humans do on the web, namely searching and filtering information. The programs can then execute transactions based on the information, potentially in an autonomous manner. The Semantic Web structure uses other technologies to enable automated search and transactions. We will discuss some of them later, in **Error! Reference source not found.**.

1.2.3 Ontology, the Semantic Web and Biomedical informatics

Biomedical informatics is a field of computer science dedicated to studying and developing computerized information systems for the health care industry. Since its early times researchers in this field recognized the importance of representing medical knowledge into information systems and various attempts in that direction were started[6] . Ontology is among the most powerful tools to encode medical knowledge formally[7].

On the other hand, the Semantic Web can provide the supporting infrastructure to enable access and use of repositories of such knowledge distributed over the future web[8]. That is why biomedical informatics is intimately connected to and invested by both the trends I sketched above; in fact, not by accident, ontology has established a solid ground in academic and industrial research and development in this arena[9].

It is worth clarifying that the present work, while stemming from the broad debate on medical ontology, does not directly participate in current efforts for using ontology in the standardization of medical data transaction or for establishing controlled biomedical nomenclatures[10]. Rather it uses ontological methods to formalize entities and relations among them, in the rather abstract domain of intentional planning. We will discuss later the details of what the research area of planning, a branch of artificial intelligence (AI), consist in. In the following chapter, we will discuss some of the issues addressed by ontological design methods.

1.2.4 Ontological theories

As a branch of research, stemming from philosophy and only recently employed by computer scientists, ontology is still evolving and different approaches are currently proposed to create ontological representations[11]. One of the main distinctions among approaches is the manner in which concepts and reality are defined. The cognitivist approach considers concepts, or ideas, in the same way as real objects, regardless of their relation to reality: a femur and a unicorn are therefore tackled in the same way in the construction of the ontology. Realists on the other hand exclude non-existing objects from their designs. Pros and cons of these methods are very much debated, but as long as correctness is guaranteed, they both generate effective and equivalent representations.

Another issue currently discussed in this area is the utilization of multiple inheritance opposed to single inheritance where all objects in the ontology inherit their attributes from a single parent class or type. Those in favor of single inheritance generally argue that chances of committing classification mistakes are less if multiple parents are not allowed. A more flexible line of conduct suggests choosing type of inheritance according to the domain to be modeled, since different domains might require different approaches. Generally, those who advocate the use of multiple inheritance insist on its ability to provide compact and easily manageable representations and to account for multiple 'views' on the objects. What, for instance, in a nosographic partition, can be classified as a *pathology,* may be considered a *diagnosis* in a more clinically oriented classification. Both these definitions can coexist in a multi-partitioned ontology, that is an ontology that uses multiple inheritance.

We will illustrate our approaches regarding these issues in the chapter on methods.

1.2.5 Ontological frameworks

Part of our study has been dedicated to the alignment of our ontology with so-called top-level ontologies. These particular types of ontologies can be defined as comprehensive philosophical models of reality, encompassing all root concepts or entities necessary to define, on lower levels, specific domains. Efforts in this area date back to Aristotle and more recently to Leibniz and other contemporary thinkers (C. S. Peirce, J. Stuart Mill among others)[12]. In contemporary studies on this matter, substantial differences exist in the envisioned categories, but generally, these are distinguished in two major classes: endurants and occurents[13] [14]. The accent is on what in the universe has a temporary existence, that is an occurent and what instead can be considered eternal or lasting for long periods of time so that is convenient and acceptable to discard its transitory nature. This approach has many desirable practical consequences, not least a crisp distinction between processes, occurrents, and objects, frequently considered endurants, which is fundamental to describe most of the phenomena taking place in the world. Classic examples of endurants are for instance computer algorithms and scientific theories or, in the physical realm, material objects such as rocks or books. Occurrents are frequently associated with the idea of processes and describe those entities such as living organisms or meteorological phenomena with limited time of existence and visible beginning and end.

In our design, goals have been considered as occurrents, while plans have been considered as endurants. We will elaborate later, the reasons of this choice.

Section 1.3. Description logic

In the previous section, we described how a given portion of knowledge or reality can be described in computable terms. However, formally represented medical knowledge is only one component of a decision-support system. What is also needed is the ability of executing inferences on the declared representation. Description logic is a powerful logic-based knowledge representation language allowing complex reasoning over formally represented domains. Simpler formalisms such as semantic networks or object-oriented design have been proved useful for clarifying a certain portion of knowledge or reality. However, these formalisms do not support more advanced types of inferences such as concepts subsumption and satisfiability. Subsumption refers to the capacity of a system to decide if, given two objects in a domain, it is always possible to establish that one of them is a subclass of the second. By 'always possible' we mean that whatever the interpretation of symbols identifying objects is, the relation still holds. More formally, we can state that a concept C is subsumed from B when the relation $C^i \subseteq B^i$, where i is the given interpretation, holds in every universe or domain T considered, or $C \subseteq_T B$. Satisfiabilty is a more general idea, which guarantees the consistency of a model. In practical terms, given a universe and an interpretation, the interpretation *satisfies* a statement on the universe if and only if it is possible to map every object in the universe with a symbol in the interpretation, or, for instance: $(a , b) : R$ iff $(a^i, b^i) \in R^i$, where a and b are two objects in the domain, R is the relationship among them and i is the interpretation.

Definition of sufficient and necessary conditions, identification of single instances of a type (i.e. the patient John Smith, not just any patient with those characteristics) and tracking entities in the knowledge bases are in fact guaranteed only by these mechanisms that deal with formal semantics.

In our representation, concepts are ontological classes existing in the domain. Different statements about these concepts are written and executed, that are checked for satisfiability and subsumption, using the description logic language. The power of this approach resides in the capability gained by intersecting ontological representation with description logic that enables, for instance, classification of objects under existing categories (classes) which leads to automatic creation of inferred objects' hierarchies. It should be understood that the reasoning power gained with the ability to automatically classify objects dramatically amplifies knowledge-based systems' capabilities.

1.3.1 Protégé, Frame based knowledge representation and the OWL language

Protégé is an open source ontology editor developed at Stanford University[15]. It is based on Java and provides an environment for knowledge representation using the frame-based approach.

1.3.1.1 Frame-based knowledge representation

Frame-based systems date at least two decades back and are one of the first attempts to formally grounding the knowledge representation activity. The main idea beyond this type of models is that a concept is defined by a frame that is connected to other frames by slots which represent relationships. In other words, if a frame *f* is in a relationship *r* to a frame *g*, then the *r* slot of *f* carries the value *g*. For example, suppose we are describing the genealogical tree in Figure 1:

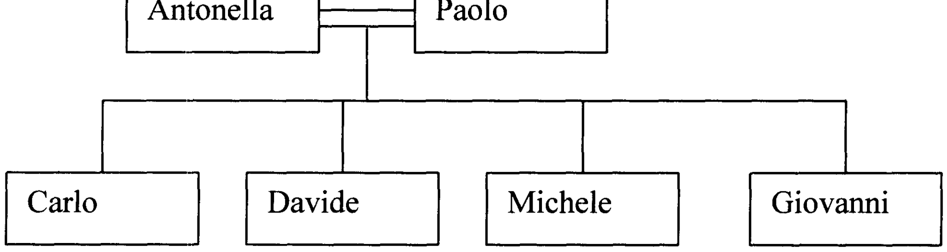

Figure 1: A frame-based representation

The frame describing Antonella might look something like:

```
Antonella:
  sex:    Female
  spouse: Paolo
  child:  (Carlo Davide Michele Giovanni)
```

where sex, spouse, and child are slots. Note that a single slot may hold several values (*e.g.* the children of Antonella).

The genealogical tree would then be described by, at least, eight frames, describing the following individuals: Antonella, Paolo, Carlo, Davide, Michele, Giovanni, Male, and Female.

1.3.1.2 OWL, The Web Ontology Language

While guaranteeing some flexibility, the frame-based approach fails to provide compact representations, particularly for complex domains and significantly limits a model's expressivity. Both these issues are addressed by description logic and the language based on it, the Web Ontology Language (OWL)[16]. Recent versions of Protégé incorporate support for OWL in the form of a plug-in [17].

The W3C describes OWL as "a language designed for use by applications that need to process the content of information instead of just presenting information to human". OWL achieves that by supporting XML, RDF, and RDF Schemas and adding to these additional vocabularies and more importantly formal semantics. Markup languages, as those just mentioned, are in fact are useful to assign definitional tags to objects in the program but are not for defining them in a computable way. OWL accomplishes this by supporting efficient description logic algorithms.

Section 1.4. Intentional Planning

M. Bratman[18] developed in 1984 the first philosophical framework for modeling artificial autonomous agents. This was then formalized by logicians in the Belief Desires

Intentions (BDI)[19] model in which three components are envisioned in the planning activity: desires, beliefs and intentions (goals).

The BDI model provides crisp distinctions and relations between the environment and the agent and between agent's desires, beliefs and intentions. Desires, for instance, are defined as not related to the context in which the agent exists (to be a rock star, sleeping instead of going to work) and therefore not directly relevant to the decision process.

In the medical domain, a desire might be the cure of a patient's illness. This may seem not just a desire but a legitimate intention, which, it is in human terms. However, I do not consider it an intention, or goal, in my ontology because it might be completely unrealistic in some unfortunate cases, for instance, in illnesses with no known cure. The need for instantiating only practically achievable intentions is clearly paramount in clinical medicine.

The second component of the BDI model, an agent's beliefs are definable in a straightforward way in computerized systems. Available data about a patient constitute what the agent believes about the patient's status. Collapsing the distinction between what the agent believes and what the state of the world is, while arguable in other contexts, does not appear to be incorrect here. Both status and knowledge of the world are assumed to be identical in our design and this assumption reasonably holds in health care institutions. Clearly, beliefs are not static but change with context. As plans are executed, either by the agent or by other external entities, the environment changes and beliefs are modified.

The final components of the BDI model are intentions that are also derived by the agent from the context and are briefly defined as commitments to a plan. Here the distinction with desires is further emphasized; no intention is such if a viable plan cannot be identified for its achievement.

Through these three components, an agent makes explicit, or gathers, the needed information and selects possibly effective plan of action to achieve his goals. For many artificial agents, selecting a plan generally consists of executing a query against a repository, frequently called plans library, in which a static representation of plans is stored. The query is formulated by analyzing the context, or scenario. There are therefore two distinguished phases of the reasoning activity: 1) the creation of the query and 2) its execution against a knowledge base. Applying this design enables a high grade of flexibility. Definitional properties of goals, plans and other relevant classes are decoupled from the plans library (the knowledge base). This design makes it possible to change the plans library as long as the schema matches the ones specified in the core ontology and can thus enable an entirely different set of applications. The high-level definitions of goals and plans will still be valid and therefore the inference capability is preserved regardless of the knowledge base used. For example, in the medical scenario, an agent may dynamically create sets of specific medical orders in response to a newly confirmed diagnosis. In other applications, an agent may schedule medical procedures according to patient's status and health care provider's goals.

1.4.1 Hierarchical Task Decomposition

The previous section described how intentions can be used to drive the selection of plans. In the section, we discuss how plans can be represented.

In most real cases, plans or actions executed by human or artificial agents consist of complex sets of tasks unfolding in a defined order to achieve the desired result. These actions can be divided into complex and primitive ones and arranged in hierarchical form. This approach to structuring plans is known as Hierarchical Task Decomposition (HTD)[20]. In this technique, plans are segmented in their atomic components and ordered in the required sequence. Each plan in this scheme, by being successfully executed, creates the necessary preconditions for the following plans. This could be explained by the following medical example (see Figure 2).

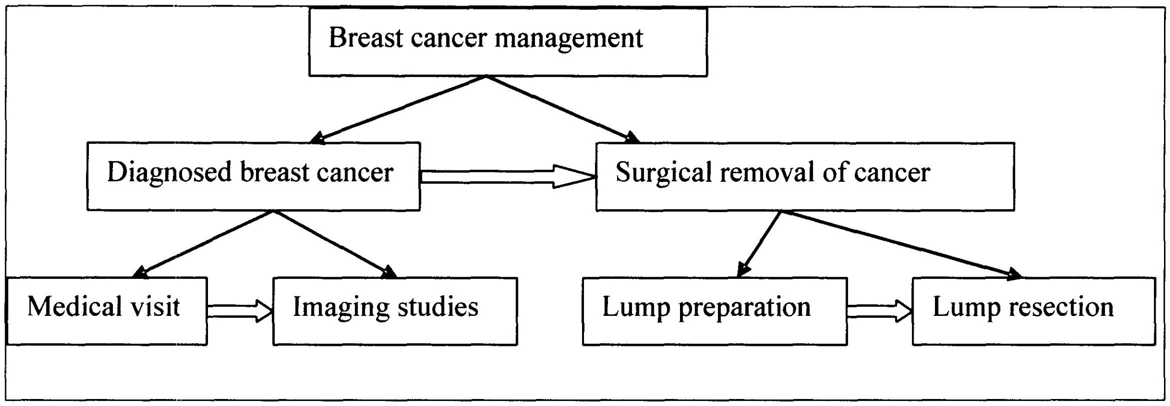

Figure 2. Hierarchical task decomposition of a medical plan

The thin solid arrows show how the plans are divided into their components while the thick, hollow, and horizontal arrows show the dependency of a plan from its predecessor, i.e. the surgical removal of a cancer, before being executed, needs to be preceded by the diagnosis of the cancer.

Every plan therefore creates the needed conditions for the following plan to begin execution; these are frequently related as plan's preconditions. Some systems[21] have features supporting the explicit specification of preconditions. We will not go into the details of most of these systems but rather discuss in the following section those experiences relevant to the medical domain and to the framework that we adopted.

Section 1.5. <u>Problem Statement</u>

The amount of information created by biomedical research has exponentially increased since the last few decades, with a further sharp acceleration since the take-off of high throughput bimolecular research, currently driving most of the advances in therapy and diagnosis. A consistent part of this knowledge has been collected and organized in ways that should enable easy access and utilization by health care providers, in the form of clinical guidelines and other so-called "evidence based" knowledge sources. In other words, this information is suitable for being incorporated into the clinical practice and utilized by physicians during their daily activities. The push in this direction has been given by both soaring costs of medicine, partially due to structural inefficiencies and variability among doctor's behaviors and the ethical concern about patients' safety in current health care institution. Medical errors have been found, in fact, to be one of the major causes of death in the western word.

For both these issues, part of the remedy is considered to reside in providing clinical decision makers with appropriate, high quality, information in a timely manner, at the site of their work.

The idea behind our work is that a more appropriate and targeted provision of medical knowledge can be automatically executed by defining, formally, in a computer interpretable way, clinicians' intentions in a given moment during the process of care. In order to achieve the intended results the system not only needs to be able to elicit and make explicit current user's intention but also has to be able to execute sophisticated inferences on these. In fact, while a goal can be easily defined, many factors affect the type of plan consequently selected or constructed to reach the goal, such as available resources, time constraints, costs and, more important the particular status of a patient. In clinical practice, there are virtually no cases with the same presentation, every patient having a particular history and manifestation of a disease. Moreover, a possibly very high number of plans, frequently more than one, are effective to achieve the same goal. Therefore, a robust method to select the most suitable plan has to be developed. We believe that an accurate formalization of medical goals using ontological design would serve this scope by grounding decision support in the operational scenario of clinical practice and responding to health care providers needs by directing and addressing their current aims.

Chapter 2. Materials and Methods

Section 2.1. Ontology Design

In the previous chapter, we briefly discussed different ways to develop ontologies and the theoretical grounds upon which different methods are based. We now explain what our approach has been with regard to ontological theories, types of inheritance and alignment to ontological frameworks.

2.1.1 Realist versus cognitivist approach

We decided to follow a realist approach in developing the ontology, or, simply stated, we represented only entities existing in reality. This choice was guided by the nature of the domain at hand, medical practice, which is involved with real objects and consists of actions performed in the physical environment. The point we want to stress is that objects, such as goals, while having no materiality, can still be represented as real. In our view, they exist as temporary mental states and warrant representation, if only for ontological reasons.

2.1.2 Multiple inheritance

We used a multi-partitioned representation within our domain, i.e., we allowed multiple inheritance. Allowing classes to have more than one parent was suggested by the intent to to avoid the redundancy of representations based on single inheritance. Nevertheless, being aware of the higher risk of incorrect classification associated with multiple inheritance, possible errors were minimized by using automated classification that is supported in description logic. This involved explicitly stating definitions in description logic syntax and then using a reasoning engine to infer alternative taxonomies. In this respect, our task was greatly simplified by the editing environment we employed: Protégé-OWL, a system that supports writing logic statements and executing inferences.

2.1.3 Mapping to ontological frameworks

Our intuition with respect to the alignment of our design to top-level frameworks is that while plans can be considered endurants, goals are better classified as occurrents. We, in fact, considered plans as entities living independently from the agents executing them and their effects. While plans are eternal, their *execution*, which has starting and ending moments, is transient. This feature is captured within the core ontology where plans are defined by attributes such as T, the plans' duration, or S, their status, which insert the execution of plans into the temporal dimension. This choice has the useful implication of allowing reasoning on concurrent and possibly conflicting plans.

Plans' structure and content may be modified over time. This is particularly true for medical plans, guidelines, which are frequently updated with recommendations based upon new discoveries. This does not alter their nature. In fact, endurants may change as long as they maintain their defining permanent status.

For the purpose of our current design, it is not imperative if a plan is deleted, for instance, due to new medical knowledge discrediting that practice. The potential importance of this aspect will be discussed in the future work section.

Goals have been defined as occurrents, and our intuition is that a goal exists only until the required conditions for its satisfaction are met. At that point, the goal dissipates and eventually other goals may be generated. In the current design, conditions are modeled in the desired clinical state attribute.

In our representation, a goal is an agent's operative state, generated by the scenario and by which the agent sorts and identifies viable plans to change the state of the world, namely the patient's clinical state in our domain.

Section 2.2. Classes definitions

Following these considerations, we operationally defined goals as ordered tuples of two components, or < DCS, P>, where:

- DCS is the desired clinical state.
- P is the goal's priority.

This last attribute was not defined further in the current version because we felt this was specific to the future application of the ontology. DCS is the clinical state a care provider would like to achieve for his patient. Our ontology defines both a diagnosis and a clinical finding as subclasses of the DCS class. In fact, these two concepts are subclasses of the same entity, specifically the available information about a patient, which refers to, but is not, his complete physical condition. This can never be fully captured due to the countless anatomical and physiological facets of a human being. In the diagnosis class, we inserted the no_diagnosis subclass to account for those cases without an established diagnosis.

The other important class in the ontology is plan. A plan, as already discussed, is an action or a set of actions that can be taken to achieve a given state of the world. A plan can be any process described by the following set of ordered attributes:

< ICS, C, G, CC, R>, where:

- ICS is the current clinical state to which the plan applies. Note that ICS is represented with the same class as the DCS in the goal's definition allowing the same entity to define the clinical scenario both for goal and for plan.
- C is the plan's cost, a general function which can be further defined according to implementation specifications.
- G is the plan's goal and takes a goal as value. This is perhaps the most important feature of plan's definition and of our model in general. By instantiating the goal attribute, the discussed commitment to a plan, which in the Bratman framework defines a goal, is enacted.
- CC is the context of care. This attributes specifies the environment in which the plan is executable. Plans may in fact be conceived for only one or few care settings such as emergency rooms or small clinical practices.
- R defines the resources needed to execute plan. Agents involved in plan execution are examples of resources as well as medical devices or facilities employed. In addition, the time required to complete the plan is considered a resource.

An overview of the above classes and relationships among them is shown in Figure 3 in the form of a class diagram in the Unified Modeling Language[22] notation.

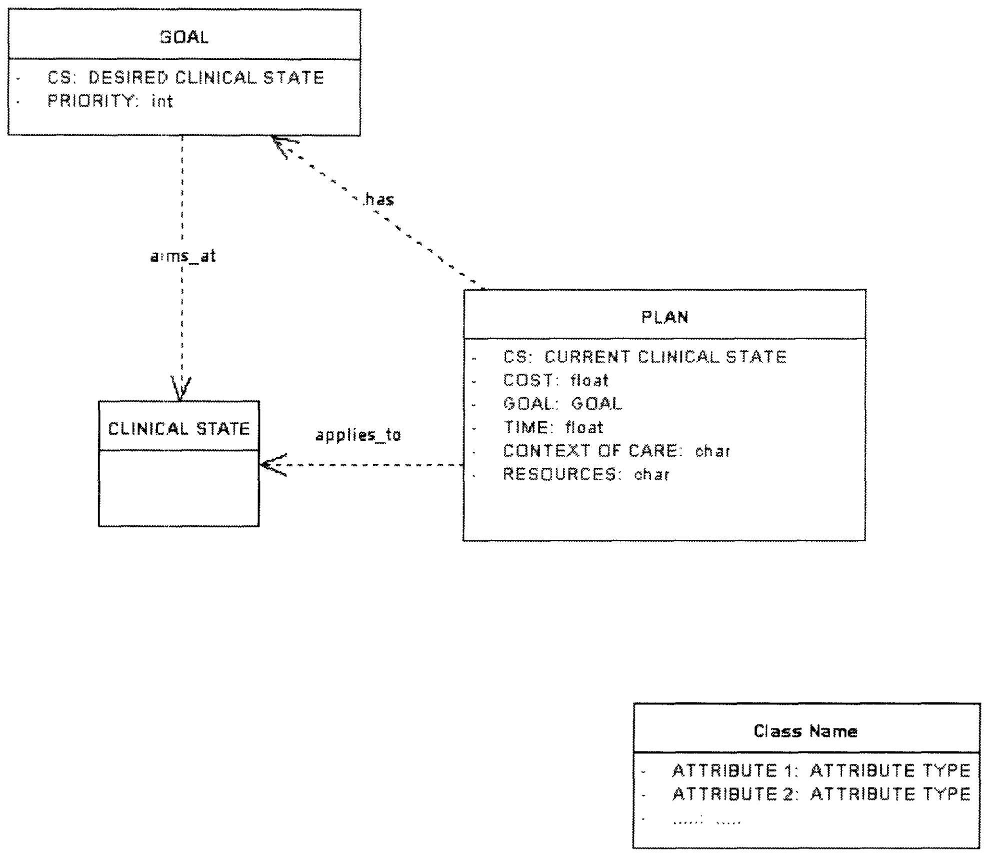

Figure 3: Overview of the ontology. Boxes describe classes in the ontology while arrows identify relationships. The box in the lower right corner clarifies a class definition.

2.2.1 General design

The current architecture consists of three OWL files containing the goals ontology, the plan library, and the test cases, respectively. These three ontologies are connected by two URIs, or universal reference identifiers. These allow importation of an ontology into another and use of its classes and properties in a distributed fashion, according to Semantic Web paradigms.

This design also allows a clear visualization of the reasoning process, which comprises our system. It distinguishes between the two phases, formulating the query and executing it, we described in Section 1.4. Definitions of goal, plan and other relevant classes are decoupled from the plan library (the knowledge base), adding flexibility to the system. The inference capability is preserved in the goals ontology, regardless of the knowledge base used.

2.2.2 The plan library

According to the HTD approach, we modeled two clinical guidelines to create the plan library: diagnosis of breast disease [23]and hypertension diagnosis and treatment [24]both from the Institute for Clinical Systems Improvement (ICSI). The HTD method involves subdividing complex plans down to their smallest components. These are called atomic plans and are defined as non-divisible action.

Examples of atomic plans from the encoded guidelines are screening mammogram, cyst aspiration and breast ultrasound exam. However, these are not to be considered atomic plans in all circumstances. In fact, a radiologist or a surgeon may want to subdivide further plans relevant to their area of expertise. We decided to limit plans granularity to what would be needed by a general practitioner.

2.2.3 Reasoning

Inference on the ontology was enabled by an external, server-based, reasoning tool called Racer[25]. This is a description logic-based inference engine interfacing through APIs with Protégé-OWL. Racer is able to directly interrogate OWL ontologies through predefined queries and through these, reason for concept satisfiabilty and subsumption. It also is able to find inconsistent concepts resulting from modeling errors and to determine a concept's parents and children.

We exploited these capabilities to simulate an information retrieval process in which the retrieved information consisted of plans, i.e. parts of modeled guidelines. We performed the information retrieval by generating classes describing clinical cases at various stages of disease. A general schema of the current architecture is displayed in Figure 4 . The diagram shows all different stages of system's use. Some objects in the figure are instances of ontology's classes (denoted by underlined names proceeded by colons). The process starts with a clinician formulating (instantiating) a clinical goal after having assessed the patient's clinical state (CS). This is shown by the identify relationship in the diagram. Following this action, other relevant classes are instantiated by retrieving their attributes' values (source of this information is deemed the hospital information system). This represents the query formulation task we discussed earlier. At this point, the inference engine executes the query against the plans library. This results in the selection of one or more applicable plans and their provision to the user. The results from the simulation are described in the next chapter.

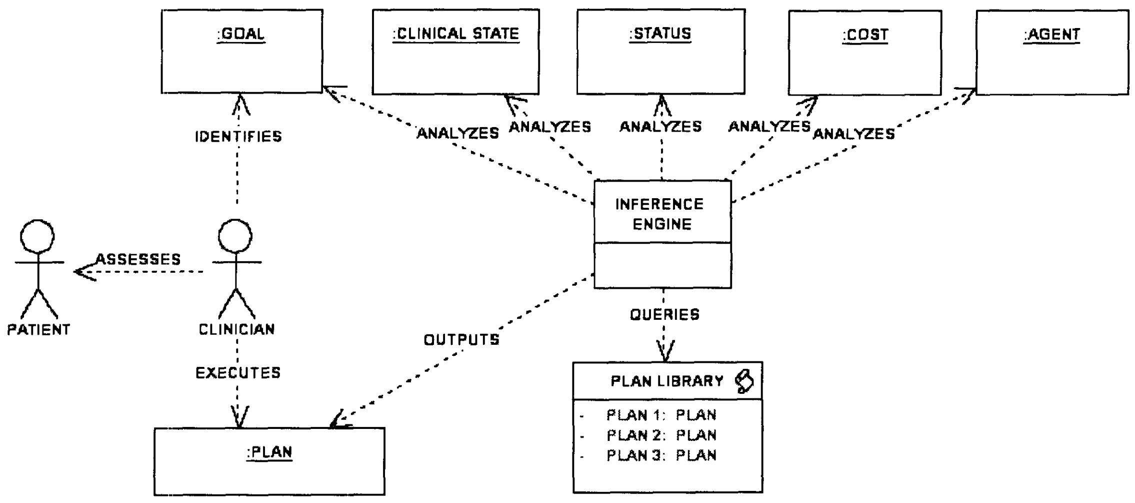

Figure 4: General architecture of the simulation.

Chapter 3. Testing the ontology

Our system's capabilities were tested by simulated retrieval of plans from the plan library for nine medical test-cases. These cases were represented in the form of classes in OWL. The classification was performed by Racer. The structure of inferred hierarchies allowed us to verify the classification's correctness. The simulation envisioned a usage scenario involving a health care provider, interested in obtaining recommendations for managing a particular patient. The intention was to provide highly focused and tailored recommendations, namely the atomic plans we already described in previous sections. In fact, we wanted to support the care process through the suggestion of directly executable clinical actions, not just general lines of conduct to be further interpreted by the user.

We created nine classes to represent patients in various clinical stages. These nine classes were subclasses of plan. Attributes, such as goal and clinical scenario were used in their definition along with other features such as age and gender. Attributes were set to be necessary and sufficient conditions in the test-cases classes. This enabled matching plans, in the plans library, with utility classes' based on matching attributes values and their subsequent classification as subclasses of the test cases. The key idea in this approach is that the inference engine would retrieve only those plans with characteristics matching patients' attributes and specified goals. A brief description of the nine utility classes is provided below.

Table 1: medical test-case, brief clinical description

Case #	Clinical Presentation
1	no breast-related or other symptoms, eligible for screening mammogram
2	no health issues, not eligible for screening mammogram
3	palpable breast mass present, no other symptoms
4	possible diagnosis of breast cancer, residual mass after cyst aspiration
5	established diagnosis of breast cancer, imaging studies completed
6	high blood pressure values at first measurement, no other symptoms

7	high blood pressure values at repeated measurements
8	diagnosed secondary hypertension
9	controlled primary hypertension, high blood glucose

The plan library currently containing 32 atomic plans taken from the two mentioned clinical guidelines is shown below.

Table 2: The Plans Libray. Atomic plans are indicated in italic

- General Plans
    - Diagnosis and Assessment
        - Physical Exams
            - Inspection
            - Palpation
            - Percussion
            - Auscultation
        - Medical History
            - Personal History
            - Familiar History
            - Social History
    - Disease Management
        - Refer to appropriate guideline
        - Referral to a specialist

- Plans for Breast Disease
  - Plans for Breast Disease Diagnosis
    - Imaging Studies
      - Mammography
      - Ultrasounds
      - CT scan
      - MRI
    - Laboratory Tests and diagnostic procedures
      - Cyst Aspiration
      - Ultrasound Guided Cyst Aspiration
    - Screening Tests
      - Screening Mammogram
    - Medical History
      - Personal Medical History
        - ask about m.h. of ductal hyperplasia with atypia
        - ask about m.h. of ductal hyperplasia without atypia
      - Family Medical History
        - ask about first degree relatives with breast disease
- ask about second degree relatives with breast disease
- ask about familiar history of other neoplasm
      - Physical Exam
        - Breast Inspection
        - Breast Palpation
        - Lymphnodes Palpation
  - Plans for Breast Cancer Treatment
    - Surgical resection
    - Chemotherapy
    - Radiotherapy
- Plans for Hypertension
  - Plans for Diagnosis of Hypertension
    - Blood Pressure Measurement
    - Repeated Blood Pressure Measurement
    - 12-lead electrocardiogram (ECG)
    - Urinalysis
    - Blood glucose
    - Serum electrolytes
    - Creatinine

Note that these plans do not represent the entire set of guidelines' recommendations. A limited number of recommendations from these guidelines were encoded for purposes of this test.

A set of general plans has been created, as can be seen in the top section of the hierarchy in the above table,. Every guideline assumes many clinical tasks, such as general physical exam or medical history, to be part of the clinical process without making this explicit. We consider the possibility of adding other types of recommendations, external to the guideline, a useful feature, as we will explain later.

We already described the definition of plans. The screening mammogram plan, for instance, is formally defined as

Plan Λ

∃ hasInitialClinicalState ((NoPalpableMass ⊔ NoDiagnosis) ⊓ moreThan35years))

∃ hasGoal DiagnoseBreastDisease

Corresponding to this plan, Case number 1 is defined as female patient older than 35 years with no palpable mass. As shown in **Error! Reference source not found.**, the inference engine classifies the screening mammogram under the Class, Case number 1.

Table 3: Test-cases and the results of plans classification

Case N.1	
	Screening Mammogram
Case N.2	
	No Plans Retrieved
Case N.3	
	ask about m.h. of ductal hyperplasia with atypia
	ask about m.h. of ductal hyperplasia without atypia
	ask about first degree relatives with breast disease
	ask about second degree relatives with breast disease
	ask about familiar history of other neoplasm

Cyst Aspiration

Case N.4

Referral to specialist

Ultrasound Guided Cyst Aspiration

Case N.5

Referral to a specialist

Surgical resection

Case N.6

Repeated Blood Pressure Measurement

Case N.7

12-lead electrocardiogram (ECG)

Urinalysis

Blood glucose

Hematocrit

Serum electrolytes

Creatinine

Case N.8

Referral to specialist

Refer to appropriate guideline

Case N.9

Refer to appropriate guideline

Chapter 4. Discussion

Section 4.1. Goals in clinical decision making

Studies of care providers' strategy for decision-making suggest the importance of formalizing their intentions, in order to intervene appropriately with decision-support actions. According to current research in this field[26] [27], clinicians iteratively generate and refine their clinical hypotheses that incrementally restrict their choice of interventions until a decision is finalized about what action should be executed. The search for a plan is limited in scope by the clinical scenario. There is wide consensus that the scenario alone is not a sufficient component in a computerized reasoning system[28]. For this reason, our model focuses also on goals' definition.

To explain why the clinical scenario cannot alone fully direct the search for viable plans, the following example may be considered: in a patient with a serious cardiac deficit and who is obese too, the clinical scenario would consist of two clinical conditions. Providing the physician, primarily engaged in the treatment of cardiac insufficiency, with dietary recommendations would be probably inadequate because of the immediate danger deriving from the cardiac condition. Specifying that the current goal is to restore patient's normal hemodynamics would lead to the selection of plans in accordance with the situation. An important feature of clinical goals is their priority (P in our definition). The value of the priority attribute is instrumental for establishing precedence among conflicting or concurring plans. In the example we presented above (a serious cardiac deficit in an obese person), this attribute can be used to rank relevant plans. In fact, recommendations about weight reduction may still be applicable and therefore provided, as long as their provision does not interfere with clinician main focus, the cardiac deficit. A recent study describes highly undesired consequences of applying multiple care protocols in absence of methods to filter recommendations in respect to their relevance and applicability[29].

Section 4.2. Medical plans and clinical guidelines

More than a decade ago, a vast movement started in the medical informatics community, with the intent of integrating clinical guidelines in health care practice, by means of computerized systems [30] [31] [32]. The aim was to drive clinical medicine toward higher standards of quality and safety for patients. As a result, various formalisms to encode concepts and logical structure used by guidelines into computer programs were developed.

Typical guidelines' concepts are "decision" and "task", among others. These objects are generally represented as nodes of a flowchart. A decision node is a state where the care provider decides the best course of action given the clinical scenario. An action node is instead a state where an atomic clinical process is executed. The entire sequence of states finally encodes the guideline's logical structure.

Examples of these models are Asbru, collaboratively developed at Ben Gurion University and the Vienna University of Technology, GLIF, developed by groups at Columbia, Stanford and Harvard Universities, Pro*Forma*, of the Advanced Computation Laboratory of Cancer Research, UK and Guide, developed at the University of Pavia.

These projects attempted a comprehensive representation of all guidelines' aspects to make them readable and executable by a computer. Here, we want to contrast only one aspect, among others, they share with our design, namely the mechanism designed to select an action, or plan, given the clinical state and provider's intentions. Authors of the above-mentioned models refer to this feature as decision-making or control flow, none of these terms really describing our view on this issue. There is a fundamentally different approach in our design: rather then representing the actual guideline's structure, or flow, we decided to decompose it into its atomic tasks and define each of these separately. Tasks, i.e. atomic plans, inherit their attributes from the plan class and are instantiated in the plan library, i.e. their attributes take on specific values. The inference engine then identifies what plan can be selected given the context, or in other words, it infers the guideline's logical structure. This approach keeps the guideline's inner logic implicit and creates independent knowledge bits (plans). These two features are instrumental for allowing the creation of easily manageable and expandable knowledge bases containing heterogeneous content. From the modularity, created by modeling plans on the atomic level, results the simple structure of the knowledge base. This in turn allows merging different guidelines and eventually other types of medical knowledge, such as clinical orders, into a single repository. A medication prescription can, in fact, be viewed as a medical plan and modeled using the approach we described for guidelines' recommendations. The system will then be able to provide the user not only with the guideline recommendation but also with the order to be entered in that clinical scenario.

Section 4.3. Modeling Plans

As we already explained, plans' preconditions describe the state of the world that needs to be realized before a plan can be executed. In the medical domain, preconditions are very important. They can represent eligibility criteria for a clinical protocol, indications for a therapy, or the required clinical conditions for a patient to be discharged. Frequently preconditions are realized by the execution of other plans. Rather then just defining plans preconditions in logical statements, we wanted to experiment a more articulated way of representing plans and their preconditions using OWL-S.

OWL-S [33]is an ontology of web services. We attempted to build an ontology of plans using the OWL-S plug-in for Protégé. The use of OWL-S to design other types of processes has not been well established. The decision was taken considering that Web Services (WS) share much of their representation with that of real-world plans. WS generalize an automated customer-provider transaction and are defined by attributes such as precondition, inputs, participants, results and others specifying, for instance, execution patterns. The OWL-S plug-in provides dedicated tools to specify such attributes.

Our strategy was nevertheless not entirely successful. The main reason for this outcome is that in order to model plans in OWL-S, their definition needs to be aligned with the Web Services ontology that is part of the OWL-S plug-in. This ontology is narrowly focused on web services and its integration with other ontologies proved to be a non-trivial task. We imagine this to be prohibitive for other non-web-services ontologies too. A different approach would be to configure the plan library entirely in OWL-S (so what did you do with OWL-S). We think that this might not be entirely appropriate and lead to design issues due, again, to the very specific domain of the Web Services ontology which may not be easily expanded to other domains. Our general impression is that OWL-S, while appearing as a very promising approach for future developments in this area, can currently support representation of very limited number of non-web-services plans.

We want now describe how the other systems we mentioned in the previous section addressed this same issue.

The GLIF model specifies goals as text strings [34]. These are used by users for searching. ProForma, Asbru and Guide support formal specifications of goals and use it to influence selection of recommendations. In Asbru, goals and plans are coupled in "temporal patterns of provider actions and patient states, at different levels of abstraction, that should be maintained, achieved, or avoided"[35]. Goals are also classified in four categories according to their granularity (intermediate goals or overall goals) and to their procedural component (state goals versus action goals). Furthermore, goals are designed as optional features. This contrasts with our view of the decision-support activity, which, as explained, should not disregard the clinician's intentions.

In ProForma[36], goals are specified as tasks' attributes. Goals representation is made of two components, the verb and the object (e.g. goal = remove(verb) : foreign body (object)).

Our approach contrasts with these others because of the strong focus on goals' analysis and the resulting architecture, which uses them for plans selection. While in the designs discussed above, intentions are secondary or optional attributes, a decision-making system based on our ontology would use goal as starting point for recommendation provision.

We provided in a previous study [37] a richer definition of goals as a set of five attributes. The first attribute of that definition, the initial state or context in which the goal applies, is in this version, a plan's feature, ICS. We wanted to highlight in this study, how a goal definition identifies a target in the future rather than acknowledging the current state of the world, according to the ontological framework we established. Goal in our design can be seen as modifiers, or filters, in the selection of plans, which is initiated by the analysis of the initial clinical state (ICS) and directed toward the desired clinical state by the goal. Other two attributes of the previous definition, the intentional verb and the target, have been collapsed into the desired clinical state attribute. We deemed this as more agile approach serving better our purpose of demonstrating overall system's reasoning capabilities. The utility of using verb/object, or verb/target, constructs in goal definition, something that most of the systems do, is related to the possibility of controlling complex plans cycles. If, for instance, a plan to decrease blood pressure is activated, its stopping condition will be the achievement of normal blood pressure values. The goal's verb component, *decrease,* can be used as a parameter to infer such condition. These types of inferences were not in the scope of this study and we opted for a more compact representation.

Section 4.4. Limitations

Currently, the plan's definition is not developed sufficiently to support reasoning and execution in operational clinical applications. Temporal patterns of plans execution, such as repetitions at time intervals are, for instance, not tackled by plan's current definition. Plans cycles and intervals between them are considered an important feature for a clinical decision–support tool. OWL-S, that we described earlier, provides dedicated mechanisms to define a plan's cycles and their starting and ending conditions. Limitation of OWL-S's use has been highlighted.

The need to specify goals explicitly might also be considered an issue from a usability perspective. We think that this could be minimized by providing appropriate interface features or embedding learning algorithms as we discussed in the previous section.

A limitation that we feel is important is the inability to enter and conduct inference over numeric attributes in expressions (e.g., body_temperature > 104) in OWL. Quantitative reasoning is clearly paramount in medicine. The Protégé development team is currently working on an extension of the OWL plug-in to add this capability.

Section 4.5. Future work

The future development of our system will focus, among other things, on the assessment of goals from real clinicians. That could be achieved by providing system's users with tools to assert goals directly in the system. For example, the user could select goals from a list. Another method, with lower impact on usability, would be to utilize learning algorithms, such as logistic models, neural or Bayesian networks. These would be trained by the user or by memorized usage patterns to predict users' goals.

Another aspect that still needs to be addressed is the specification of plans preconditions. An optimal development of such feature should provide tools to specify logical and temporal relations.

Specification of plans preconditions and complex execution patterns will also be an aspect to develop in future efforts.

One relevant aspect in medical knowledge management is methods to keep track of subsequent modification in the content. As for clinical guidelines, this is achieved by indicating information about versions and revisions applied to the text since its first creation. In a future development of the plans library, that aspect should be considered.

Section 4.6. Conclusion

We developed and tested an ontology for plans, which, we believe, offers a promising approach to develop successful implementations of clinical decision-support systems.
Important features of our approach are the possibility of creating and managing repositories of heterogeneous medical knowledge, the ability of representing complex clinical scenarios and of reasoning over them, using clinicians' intentions to identify plans of actions.
Formal definition of goals has been demonstrated instrumental to allow our system to execute correctly the desired reasoning processes.
Limitations in our approach were due partly to lack of advanced development tools rather than to theoretical issues. Because of the rapid evolution in this area of research, we believe these tools will soon be available. That will provide us with the needed support to develop our approach further.

References

1. Bach, J., What Software Reality Is Really About. Software Realities, December 1999

2. Shapiro, S., Splitting the Difference: The Historical Necessity of Synthesis in Software Engineering. IEEE Annals of the History of Computing, December 1997

3. Brooks, F.P., No Silver Bullet. Essence and Accidents of Software Engineering. Information Processing, March 1987.

4. Shaw, D.G., An Introduction to Software Architecture, in Advances in Software Engineering and Knowledge Engineering, Volume I,, V.A.a. G.Tortora, Editor. January 1994, World Scientific Publishing Company: New Jersey.

5. Tim Berners-Lee, J.H., The Semantic Web. Scientific American, 2001.

6. Randall Davis, Bruce G. Buchanan, and Edward H. Shortliffe, Production Rules as a Representation for a Knowledge-Based Consultation Program. Artificial Intelligence, 8: 15-45 (1977)

7. Nicola Guarino, Formal Ontology and Information Systems. Proceedings of FOIS'98, Trento, Italy, 6-8 June 1998. Amsterdam, IOS Press, pp. 3-15.

8. OWL - Web Ontology Language Reference. http://www.w3.org/TR/owl-ref

9. Schulz, S., Hahn, U. Ontological foundations of biological continuants. In: Formal Ontology in Information Systems. Proceedings of the 3rd International Conference - FOIS 2004. Turin, Italy, November 4-6, 2004.

10. Werner Ceusters, Barry Smith, Jim Flanagana. Ontology and Medical Terminology: Why Description Logics Are Not Enough. Towards an Electronic Patient Record (TEPR 2003), San Antonio 10-14 May 2003.

11. Six Questions on the Construction of Ontologies in Biomedicine Anand Kumar et al. http://www.uni-leipzig.de/~akumar/OntologyinDemocracy.pdf

12. John F. Sowa, Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks Cole Publishing Co., Pacific Grove, CA, ©2000.

13. BFO/MedO: Basic Formal Ontology and Medical Ontology Draft 0.0006 (13. July 03)

14. Domenico M. Pisanelli, Aldo Gangemi, Massimo Battaglia, Carola Catenacci. Coping with Medical Polysemy in the Semantic Web: the Role of Ontologies, MEDINFO 2004 M. Fieschi et al. (Eds), Amsterdam: IOS Press

15. Knublauch, Ray W. Fergerson, Natalya F. Noy and Mark A. Musen, The Prot´eg´e OWL Plugin: An Open Development Environment for Semantic Web Applications Holger http://protege.stanford.edu/plugins/owl/publications/ISWC2004-protege-owl.pdf

16. World Wide Web Consortium. OWL Web Ontology Language Semantics and Abstract Syntax. W3C Recommendation 10 Feb, 2004.

17. Knublauch, Ray W. Fergerson, Natalya F. Noy and Mark A. Musen, The Prot´eg´e OWL Plugin: An Open Development Environment for Semantic Web Applications

18. Bratman, M. 1984. Two faces of intention. Philosophical Review 93: 375–405.

19. Georgeff M., Pell B., Pollack M., The Belief-Desire-Intention Model of Agency Editors: Milind Tambe, MichaelWooldridge

20. Erol K., Hendler J., Nau D.S., Semantics for Hierarchical, Task-Network Planning Technical report CS-TR-3239, UMIACS-TR-94-31.

21. Fox J. and Das S. Safe and Sound: Artificial Intelligence in Hazardous Applications. Jointly published by the AAAI, Menlo Park, CA, and MIT Press, Cambridge Mass., 2000. ISBN 0-262-06211-9

22. Rumbaugh J., Jacobson I., Booch G., The Unified Modeling Language reference manual Addison-Wesley Longman Ltd., Essex, UK

23. Institute for Clinical Systems Improvement, Diagnosis of breast disease. 48 p., November 2003

24. Institute for Clinical Systems Improvement, Hypertension diagnosis and treatment. February 2004

25. Haarslev V., Moller R., Racer: A Core Inference Engine for the Semantic Web http://www.sts.tu-harburg.de/~r.f.moeller/racer/papers/2003/HaMo03e.pdf

26. Kee F., Jenkins J., McIlwaine S., Chris Patterson, Sloan Harper, and Michael Shields, Fast and Frugal Models of Clinical Judgment in Novice and Expert Physicians. Med Decis Making, July/August 2003; 23: 293 - 300.

27 Redelmeier D., Shafir E., and Aujla P. S., The Beguiling Pursuit of More Information. Med Decis Making, September/October 2001; 21: 374 - 379.

28. Kumar A. Clinical Guidelines as Plans: An Ontological Theory et al Ontologies in Medicine. Volume 102 Studies in Health Technology and Informatics. IOS Press, 2004.

29. Boyd C. M., Clinical Practice Guidelines and Quality of Care for Older Patients With Multiple Comorbid Diseases Implications for Pay for Performance, JAMA, August 10, 2005—Vol 294, No. 6

30. Peleg M., GLIF3: The Evolution of a Guideline Representation Format, In: Proc AMIA Symp; 1999. p. 701-705.

31. Shahar Y, Miksch S, Johnson P. The Asgaard Project: A Task-Specific Framework for the Application and Critiquing of Time-Oriented Clinical Guidelines. Artificial Intelligence in Medicine 1998;14:29-51.

32. Humber M., Butterworth H., Fox J., Thomson R., Medical Decision Support via the Internet: PROforma and Solo. V. Patel et al. (Eds), Amsterdam: IOS Press, 2001 IMIA 464

33. Sirin E., Parsia B., Planning for Semantic Web Services. http://www.mindswap.org/papers/SWS-ISWC04.pdf

34. Peleg M, et al. Comparing Computer-interpretable Guideline Models: A Case-study Approach. JAMIA. 2003;10:52-68

35. Shahar Y, Miksch S, Johnson P., The Asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines. Artif Intell Med. 1998 Sep-Oct;14(1-2):29-51.

36. Fox J, Johns N, Lyons C, Rahmanzadeh A, Thomson R, Wilson, .PROforma: a general technology for clinical decision support systems. Comput Methods Programs Biomed. 1997 Sep;54(1-2):59-67.

37. Hashmi N, Boxwala, A, Zaccagnini, D, Fox, J "Formal Representation of Medical Goals for Medical Guidelines" Proceedings, Medinfo 2003.