An architecture for building intelligent agents applied in health care based on Answer Set Programming

Juan Carlos Nieves and Ulises Cortés
Universitat Politècnica de Catalunya
Software Department
c/Jordi Girona 1-3, E08034, Barcelona, Spain
{jcnieves,ia}@lsi.upc.edu

Mauricio Osorio
Universidad de las Américas - Puebla
CENTIA, Sta. Catarina Mártir
Cholula, Puebla, 72820 México
osoriomauri@googlemail.com

ABSTRACT

To consider cognitive models which capture clinical reasoning may help to design medical tools whose behavior can be closer to clinicians' reasoning. In this paper we describe a general architecture for building intelligent agents applied to health care. This architecture is based on a cognitive model which captures clinical reasoning; moreover we consider answer set programming for performing medical reasoning.

1. INTRODUCTION

It is well-known that decision making is the medical doctor's major activity. In fact, a lot of research has been performed to study how doctors make decisions [14]. The complexity of clinical reasoning has been demonstrated by studies covering diverse medical tasks, including decision making, identification of clinical errors, and comprehension of clinical information. These studies have shown that the type of reasoning and strategies vary among clinicians; especially as in function of expertise, knowledge and problem difficulty [2]. In Artificial Intelligence (AI), several approaches for representing medical reasoning have been developed and used in the design and implementation of decision support systems *e.g.,* bayesian probabilistic methods, case-based reasoners, among others. However most of these methods uncover some of the actual complexities of clinician's reasoning [2]. Nowadays a very active line of research is devoted to study how to avoid wrong clinical decision making in diagnosis.

J. F. Arocha *et al.*in [2] remark that the application of formal methods for the representation of clinical reasoning as used by clinicians may become an important consideration in the design of decision support tools that match the clinicians' decision processes. Based on this remark, to consider cognitive models which capture clinical reasoning may help to build medical tools whose behavior can be closer to clinicians' reasoning. As usual in the design of knowledge-based systems, one of the first steps is to decide how the medical knowledge must be structured and represented.

According to [2], in the case of diagnostic reasoning the biomedical knowledge can be described as an ontology with multiple layers of concept types and various relations inbetween that serves to describe diagnostic reasoning as a process of abstracting case information from different ontological levels. This ontology is a classification of medical knowledge for use in problem solving situations and was firstly suggested by Evans and Gadd [8]. It is worth mentioning that this classification has been used extensively in medical cognition research [2]. The classification suggested by Evans and Gadd is presented in Figure 1. This classification is composed by levels of knowledge (see Figure 1) the higher levels subsume or provide a context for the interpretation of the lower levels (*i.e.* , the interpretation of symptoms and/or observations are interpreted as a consequence of a potential disease)[1].

1. The first level is the empirium, which corresponds to the basis description of sensory data and carries no medical interpretations, such anatomical descriptions or skin color.

2. The second level is composed of observations, which are perceptual categories that serve as basic for clinical classification, and hence require medical knowledge to identify and categorize. For example, patterns of shade in a radiological image or distinguishable heart sounds, which may be imperceptible to an untrained eye or ear are interpreted as observations by physicians.

3. The third level is composed of findings, clusters of observations that are interpreted in terms of their clinical relevance, such as when shortness of breath, for example, is interpreted in the context of myocardial infarction.

4. The fourth level is composed of facets, representing sub-diagnostic categories that suggest potential diagnoses, *e.g.,* cardiovascular, and discard some other, *e.g.,* pulmonary. Facets capture patterns of findings as whole concepts. For example, categorizing a cluster of findings *e.g.,* chest pain, sweating, and faintness, as a facet, *e.g.,* cardiovascular problem, serves to explore a particular subset of diseases while discarding others.

5. The fifth level is composed of diagnoses which are clinical categories with more or less known explanatory and therapeutic models.

6. In the last level, the global complexes are described. The global complexes are the circumstances that affect a particular patient, such as particular age groups or patient characteristics that may influence a diagnosis or a management path.

With this classification, the diagnosis can be considered as a narrow-down search process in the space of possible diseases that account for the clinical manifestations. Observe

[1] The description of each level was taken from [2].

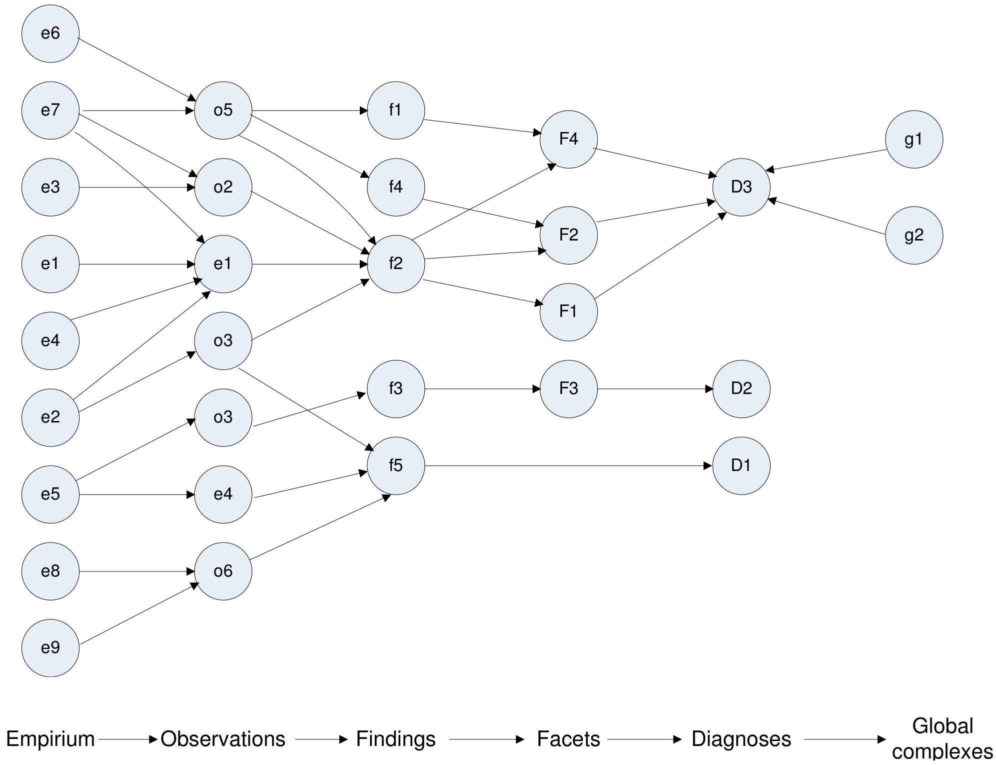

Figure 1: Ontological model for clinical problem solving.

that this classification integrates: the space of diseases (the level of facets and the level of diagnoses), the space of clinical manifestations (the level of empirium and the level of observations), and the constraints to search, such as the contextual or causality relationships between diseases and clinical manifestations (the level of findings) and the specific condition of a patient (the level of global complexes)

Based on the fact that the classification suggested by Evans and Gadd is the result of a cognitive research which captures the medical reasoning, we suggest that this structure can be considered for structuring the knowledge base of an intelligence agent applied to health care. We belief that by structuring a medical knowledge as in Figure 1, an agent can minimize the number of variables for diagnosing a disease.

Once we have identified a potential structure for modeling medical reasoning, the next question is which knowledge representation approach can be adapted in a natural form to this structure. As can be seen from Figure 1, each node of the classification can be regarded as a frame or a subset of the general medical knowledge base. Each frame can suggest a set of potential options. For instance, if there is an observation that the patient has fever, this frame can suggest at least three potential diseases: a possible infection, an inflammatory disorder and a possible cancer. One natural representation of this observation can be:

$$observation(fever).$$

$$disease(infection) \vee disease(inflammatory\_disorder) \vee disease(potential\_cancer) \leftarrow observation(fever).$$

This representation is under the syntax of logic programming (Answer Set Programming [3]).

1. The first clause expresses the fact that there is an observation that the patient has fever and
2. the second clause expresses medical knowledge. In particular, this clause suggests the three potential diseases which can cause fever in a patient *i.e.* a possible infection, a possible inflammatory disorder and a possible cancer.

In knowledge-based systems, logic is frequently taken as a language for the representation of knowledge. In fact, Lucas in [15] pointed out that logic is one of the major candidates as knowledge representation language in future-generation of knowledge-based systems. The reason for this is two-fold:

- Most other knowledge-representation languages exist in many different flavours; almost none of these languages is completely understood.
- Logic is the unifying framework for integrating knowledge-based systems and database systems.

He also remarked that although a meaningful portion of medical knowledge may be accessible to formalization in logic, for many problems types in medicine, logic will not be first language of choice [15]. Examples of such problems are *medical decision making under uncertainty* and *therapy planning*.

Since 1993 when Lucas wrote his paper, the advance in logic programming has been huge. Nowadays there are solid programming approaches which are based on logic. In fact, these approaches have representation languages which are able to capture knowledge bases of real domains. In particular, we can stress the success of Answer Set Programming (ASP).

ASP is the realization of much theoretical work on Non-monotonic Reasoning and Artificial Intelligence applications. It represents a new paradigm for logic programming that allows, using the concept of *negation as failure*, to handle problems with default knowledge and produce non-monotonic reasoning [3]. By using answer set programming, it is possible to describe a computational problem as a logic program whose answer sets correspond to the solutions of the given problem. For instance, let us consider again our observation frame which tries to suggest the causes of fever in a patient.

$$observation(fever).$$

$$disease(infection) \vee disease(inflammatory\_disorder) \vee disease(potential\_cancer) \leftarrow observation(fever).$$

This program, in terms of answer set programming, has three possible answer sets:

$$\{observation(fever), disease(cancer)\}$$

$$\{observation(fever), disease(inflammatory)\}$$

$$\{observation(fever), disease(infection)\}$$

Observe that each answer set can be considered as an entry of the third level of the knowledge structure of the Figure 1. In fact, we can regard a knowledge base as a directed graph where a node is a frame and the edges between the nodes are the answer sets of each frame.

In this paper we will show that answer set programming is a suitable approach for modeling a knowledge base architecture based on the ontological model for clinical problem solving of Figure 1. In fact, we will suggest that a medical knowledge base can be regarded as the union of small programs which we will call frames. These frames will represent a schema or a plan *w.r.t.* an empirium, an observation, a finding, a facet and a diagnosis.

The rest of paper is structured as follows: In §2, we present a short overview of the approach of answer set programming. In §3 we will define a general architecture for building intelligent agents to be applied to Health Care. This architecture is inspired in the Evans-Gadd's classification medical knowledge (see Figure 1). In §4, we comment the related work that there exists *w.r.t.* our approach. And in the last section we outline our conclusions and future work

2. ANSWER SET PROGRAMMING

Answer Set Programming (also it is called Stable Logic Programming or A-Prolog) is the realization of much theoretical work in Non-monotonic Reasoning and AI applications of Logic Programming in the last 20 years ([3, 10].

By using answer set programming, it is possible to describe a computational problem as logic program whose answer sets correspond to the solutions of the given problem (see Figure 2). For instance, an ASP program encoding a planning scenario has as many models as valid plans. This schema is similar to that underlying the application of SAT algorithms to AI, and in fact the ranges of applicability of these two techniques are similar. However, thanks to the inherent causal aspect of answer set semantics, we can represent *default assumptions*, *constraints*, and *uncertainty*

Several ASP solvers are now available, the most popular software implementations to compute answer sets are DLV [6] and SMODELS [27]. Several others can be found through the Library of Logic Programming Systems and Test Cases[2]. These systems support provably correct inferences and are at least as fast and scalable as SAT checkers. These are exciting results for the Non-Monotonic Reasoning community and they are attracting the attention of researchers from fields such as *planning* [7], *cryptography* [1], *system configuration* [28], *argumentation theory* [19], *bio-informatics* [4] .

Unlike traditional logic programming (Prolog), ASP allows us to express disjunction and "*classical*" or "*strong negation*". ASP differs from many other knowledge representation languages by its ability to represent *defaults i.e.* statements of the form "*Elements of a class C normally satisfy property P*" [10].

An answer set program is composed of set of rules, each rules being composed of a head and a body:

$$head \leftarrow body$$

Both the head and the body of a rule are sets of literals, each literal being a possibly negated atom. Contrarily to traditional logic programs, atoms are propositional rather than first-order, and they can be negated using two forms of negation: *strong negation* (denoted by $\neg$) and *negation-as-failure* (denoted by *not*). A literal is either an atom or a negated (using classical negation) atom.

In order to illustrate the syntax of ASP, let us consider the following scenario:

> Suppose that a patient suffering from certain symptoms takes a blood test, and that the results show the presence of a bacteria of a certain category in his blood. There are two types of bacteria in this category, and the blood test *does not pinpoint* whether the bacteria present in the blood is either streptococcus viridans or X. The problem is that if the bacteria is streptococcus viridans the patient have to be treated by antibiotics of large spectrum because streptococcus viridans suggests endocarditis. However, the doctor tries not to prescribe antibiotics of large spectrum, because they are harmful to the immune system. The doctor in this case must evaluate potential choices, where each potential choice has *pros* and *cons* of various strengths and incomplete information[3].

As it is common in the medical domain, we have a medical scenario where a medical decision is needed before the decision options, or the relevant information source, are fully known. In that sense, modeling assumptions (*absence of evidence*) in our knowledge base takes special relevance. Let us consider one of the doctor's believes: *Use antibiotics of large spectrum if there is not alternative treatment.* It may be expressed using *negation as failure* as:

[2]http://www.uni-koblenz.de/ag-ki/LP/lp_systems.html

[3]This example is an adaptation of Example 6 from [13]

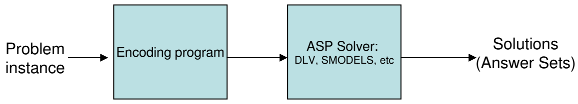

Figure 2: General schema for problem encodings in Answer Set Programming.

$$antibiotics\_large\_spectrum \leftarrow not\ alternative\_treatment$$

However, this is a dangerous way to state it: to assume that there is no available knowledge about any other alternative treatment. Instead, it would be appropriate to demand for all available explicit knowledge about whether or not there exists an alternative treatment, as could be expressed using explicit negation (strong negation):

$$antibiotics\_large\_spectrum \leftarrow \neg\ alternative\_treatment$$

The combination of *negation as failure* and *strong negation* allows for a more cautious statement as positive facts: while the rule

$$\neg antibiotics\_large\_spectrum \leftarrow alternative\_treatment$$

states that the doctor should not use antibiotics of large spectrum if there is an alternative treatment, the rule

$$\neg antibiotics\_large\_spectrum \leftarrow not\ \neg alternative\_treatment$$

states more cautiously that the doctor should not use antibiotics of large spectrum if it has not been establish that there is not another alternative treatment. We can see that the use of default negation and strong negation some times is vital for modeling *assumptions*.

A complete encoding of our medical scenario is modeled by the following logic program, denoted by Π:
The doctor knows that the patient has a bacterium of category n.

$$category\_n \leftarrow \top.$$

The category n implies two possible bacteria.

$$streptoccus\_viridans \vee bacterium\_x \leftarrow category\_n.$$

If the bacterium is *streptococcus_viridans*, then the recipient has to be treated by antibiotics of large spectrum.

$$antibiotics\_large\_spectrum \leftarrow streptococcus\_viridans.$$

If the bacteria is x, then the recipient could be treated without antibiotics of large spectrum.

$$alternative\_treatment \leftarrow bacterium\_x.$$

The doctor should not use antibiotics of large spectrum if it has not been established that there is not another alternative treatment.

$$\neg antibiotics\_large\_spectrum \leftarrow not\ \neg alternative\_treatment.$$

$$\neg alternative\_treatment \leftarrow not\ \neg antibiotics\_large\_spectrum.$$

The semantic of a program is based on its answer sets, each answer set being a set of literals. For programs not containing negation-as-failure (*not*), the semantic of a program is based on the concepts of closure and minimality:

- A program is closed under a set L of literals if the set contains at least a literal in the head of a rule whenever it contains all literals in L in its body.
- A set of literals is an answer set of a program if it is minimal (under set containment) among the ones the program is closed for.

If the program contains some literals that are negated using negation-as-failure, the semantics requires the additional concept of *reduction*.

The following definition of an answer set for extended disjunctive logic programs generalizes the definition presented in [11] and it was presented in [12]: Let P be any extended disjunctive logic program. For any subset S of the language of P[4], let P^S be the positive logic program obtained from P by deleting

(i) each rule that has a formula *not a* in its body with $a \in S$, and then

(ii) all formulæ of the form *not a* in the bodies of the remaining rules.

Clearly P^S does not contain *not* (this means that P^S is a positive logic program), hence S is called an answer set of P if and only if S is a minimal model of P^S.

An answer set program can have zero, one, or many answer sets. For instance, the program Π has two answer sets:

Answer set 1: $\{category\_n, bacterium\_x,$ $alternative\_treatment,$ $\neg antibiotics\_large\_spectrum\}$
Answer set 2: $\{category\_n, streptococcus\_viridans,$ $antibiotics\_large\_spectrum$ $\neg alternative\_treatment\}$

Observe that each answer set suggests a possible solution to our medical scenario. We accept that by considering this two answer sets, we do not have a final decision; however, the user can define some kinds of preference between answer sets. The interesting reader in defining preference between answer sets can see [5, 31, 6].

The formalization of the ASP's semantics has been studied in several non-classic logics *e.g.,* Intermediate Logic [25, 23], S5-modal logic [24], Nelson's Logic [22].

The flexibility of Answer Set's definition has permitted to regard an answer set as a *kernel* for different knowledge representation systems. For instance,

[4]The language of P is given by the set of atoms that appear in P.

- it has permitted to define a rich approach for modeling several kinds of preferences *e.g.,* weak constrains [6], strong constrains [27, 6].

- it has permitted to define several extensions for developing most expressive languages of general propose *e.g.,* Extended Ordered Disjunction [5, 31], Possibilistic Stable Models [16], possibilistic answer sets [18, 17].

- it has permitted to define systems for particular propose *e.g.,* Planning and Action [21, 7].

In ASP, it has defined a possibilistic approach [16, 18, 17] which permits to model uncertain and incomplete information that is a common feature in medical decision making. In this approach, all the rules has a necessity measure α for modeling the incomplete states of the knowledge base. The uncertain value is determined by the expert proving the knowledge base. For instance, let us consider the disjunctive clause which was presented in the introduction:

$observation(fever).$

$$disease(infection) \vee disease(inflammatory\_disorder) \vee disease(potential\_cancer) \leftarrow observation(fever).$$

As commented, the intended meaning of the first clause is that there is an observation that a patient has fever and the second one try to express the possible diseases which can cause it. We can see three possible causes of the fever: infection, inflammatory disorder and cancer. Usually, these diseases are not considered with the same possibility as a cause of fever. In fact, one can say that a possible infection is most likely as a cause of fever than a possible inflammatory disorder. Moreover cancer is the lesser possible disease as a cause of fever. Observe that this uncertainty is not captured by the actual representation. Now let us consider the following possibilistic representation:

$certain : observation(fever).$

$likely : disease(infection) \leftarrow observation(fever).$

$$maybe : disease(inflammatory\_disorder) \leftarrow observation(fever).$$

$$unlikely : disease(potential\_cancer) \leftarrow observation(fever).$$

By assuming that the relation $A \leq B$ which means that A is less possible that B and it is true that $unlikely \leq maybe$, $maybe \leq likely$ and $likely \leq certain$, we can say that an infection is most likely than an inflammatory disorder as a cause of fever. In general, we believe that this approach is a suitable approach modeling uncertain information in medical decision making. It is worth mentioning that it is also defined an approach for performing planning in terms of possibilistic logic programs [20].

Answer set programming has also influenced the development of multi-agent systems based on ASP. For instance in [30], a logic programming agent system was proposed that allows to represent the communication between decision-makers in order to come to a conclusion. The communication of the agents in De Vos and Vermeir's approach is performed by passing answer sets. In that approach is assumed that the agents are rational.

3. DIAGNOSTIC-ASP AGENT

In this section, we will define a general architecture for build intelligent agents to be applied to the health care field.

Decision making by health care professionals is often complicated by the need to integrate ill-structured, uncertain, and potentially conflicting information from several sources [14]. According to Kushniruk in [14], in recent years it has become increasingly accepted that in order to build information systems that support complex decision making it will be necessary to more fully understand human decision-making process.

In the process of designing the knowledge-architecture of an agent which main task will be to perform medical reasoning, there are some important observations that must be considered. According to J. F. Arocha *et al.*in [2]:

- Cognitive research has shown that the solution strategies and the types of inferences used during clinical problem solving are as a function of domain-specific prior knowledge that a person possesses and more specifically, of the quality and organization of such knowledge into adaptable and meaningful schemata or frames. Arocha *et al.*regard a frame or schemata to refer to learned knowledge structures in clinicians's knowledge-base that allow them to identify prototypical or familiar clinical patient problems in an *efficient manner.*

By considering this observation, we can consider the agent's medical knowledge as the union of frames such that each frame represents a schema or a plan *w.r.t.* an empirium, an observation, a finding, a facet and a diagnosis. In terms of answer set programming, a frame can be regarded as a small specification (an answer set program) with a particular propose.

Once we have divided the agent's knowledge base in small parts, we can organize these programs in layers. These layers will be exactly the layers which where suggested by Evans and Gadd (see Figure 1) plus a layer of planning. Hence, the agent's knowledge base will be captured by seven subclasses of programs:

1. Answer set programs which capture empirium;
2. Answer set programs which capture observations;
3. Answer set programs which capture findings;
4. Answer set programs which capture Facets;
5. Answer set programs which capture Diagnoses;
6. Answer set programs which capture global complexes, and
7. Answer set programs which capture plans of actions.

We want to remark that each answer set program is a frame with the same Arocha *et al*'s idea as shown in [2]. The relationship between each layer will be defined by the answer sets of each frame/answer set program (see Figure 3). A *diagnostic-ASP Agent* will be an agent which has its knowledge base structured as in Figure 3.

Observe that in the classification that we suggest there is a new layer of information which will be plans of actions. These plans of actions are related to the diagnoses. Based on the fact that usually for each diagnosis in the medical

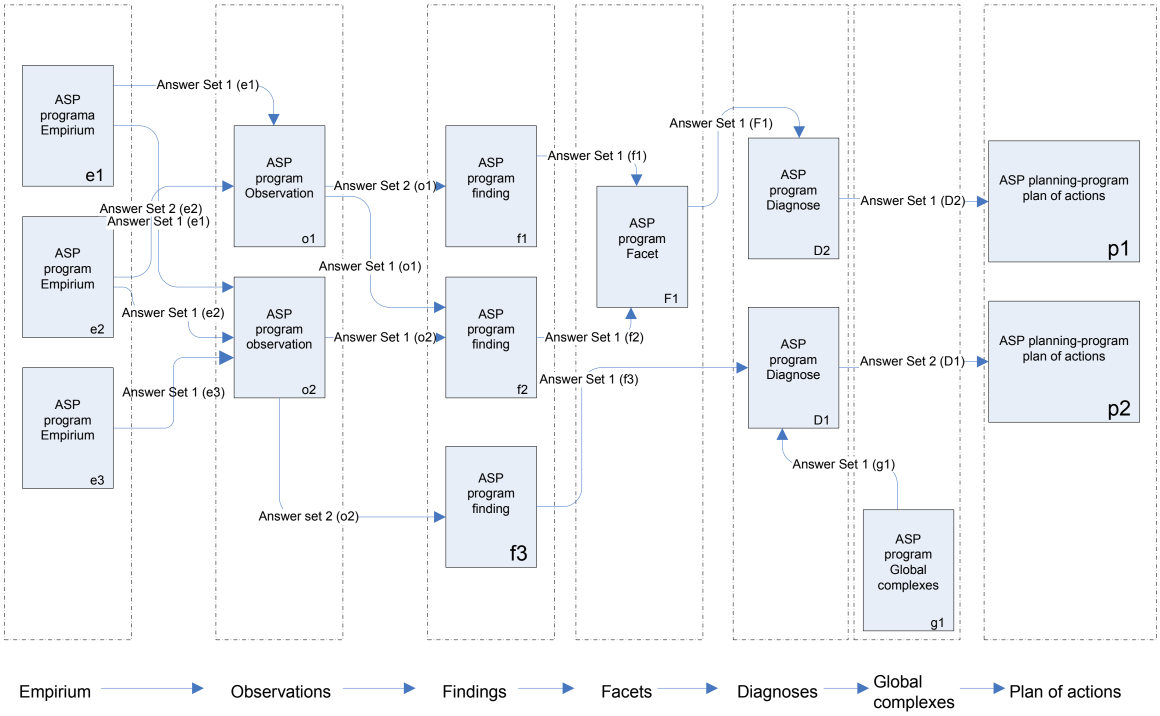

Figure 3: An architecture for building intelligent agents applied to health case based on Answer Set Programs.

domain there is a protocol of actions, we will assume that for each diagnosis there is at least a plan of action.

Some properties that we can remark of the architecture of Figure 3 are:

1. The number of variables for diagnosing a disease are minimized. It is worth mentioning that according to Arocha in [2], an efficient clinical reasoning about a disease requires to minimize the number of variables that must be held in memory in order to decrease cognitive load.

2. The explanation of a diagnosis can be given in several levels of generality. For instance, an explanation of a diagnosis can be constructed by considering just the empirium and the observations for an inexpert in medicine; however also it can be constructed a really technical explanation based on the findings and the facets. In [2], it is commented that cognitive research in medicine has shown that people generate representations of clinical cases at several levels of generality from very specific (*e.g.*, as is often the case with medical students) and very general (*e.g.*, as is true of expert clinicians).

3. The medical reasoning can be regarded as a search in a graph where the frames represent nodes and the answer sets represent the edges between the nodes.

4. The explanations of a diagnosis are coherent. The coherence is determined by the path that is constructed in the process of diagnosing a disease. Usually two form of explanatory coherence are distinguishes: local and global. The local coherence of a diagnosis is determined by the consistency or inconsistency of a frame/ASP-Program. A frame can be considered coherent/consistent when it has answer sets otherwise it is incoherent/inconsistent. The global coherence of a diagnosis is the inter-relationship among coherent frames. An explanation that exhibits local coherence, without global coherence, would include isolated components of the problem that are not explicitly linked to the rest of the explanation. According to [2], cognitive research in medicine has shown that global coherence is more common of expert clinicians, while local coherence is more often observed in less-than-experts clinicians.

It is worth mentioning that in order to implement a *diagnostic - ASP Agent*, one can take advantage of the answer set solvers that have been developed. For instance, DLV-system [6] is a deductive database system, based on disjunctive logic programming, which offers front-ends to several advanced knowledge reasoning formalisms. One interesting feature of DLV is that DLV provides an interface to database systems via ODBC (Open Database Connectivity)[5]. This means that one can consider medical database systems for constructing the medical-knowledge base of a diagnostic-ASP agent. The integration of medical database systems and knowledge based systems was remarked as one of the most important challenges in medical informatics in [15]. Another interesting feature of DLV-system which is relevant for our agent-architecture is that DLV-system has a front-end for planning under answer set programs [7]. This means that

[5] For Unix-like systems, this is achieved by using unixODBC, while for Windows systems Microsoft ODBC is used.

the last layer of our argent-architecture can be constructed in DLV-system. As a final comment *w.r.t.* DLV-system, we want to comment that DLV has a JAVA Wrapper [26] for incorporating answer set programs inside Object-Oriented programs. This suggests that we can use any Multi-Agent System platform built under JAVA, for managing the interactions of diagnostic-ASP-agents.

4. RELATED WORK

The first work that we want to comment is the work of Arocha *et al.*in [2], in fact this work was the inspiration of the ideas presented here. The first and main similarity between our approach and Arocha *et al.*'s approach is that both approaches classify the medical information according to the Evans and Gadd's classification (see Figure 1). However, Arocha *et al.*represent the information just by propositions and then they identify relations between the propositions. In our case, we adopt ASP's syntax for representing the medical information. This means that we use propositions and infer rules. For the inference of medical diagnoses, Arocha *et al.*use semantics network. As we seen in the paper, for the inference of medical diagnoses, we use answer set semantics. Another difference, it is that we add a layer for generating plans of actions based on a diagnosis.

Another important work in terms of agent theory which is close related to our approach is the domino agent model [9]. *The domino agent model* is a conceptual model for constructing rational agents (see Figure 4). This model can be thought as depicting the elements of a process in which an agent can respond reactively and purposefully to situations and events. The nodes of the domino can be viewed as data of various kinds, while the arrows are *inferences functions.* Inference mechanisms derive data of the type at the head of the arrow based on data of the type at the tail together with field-specific and field-independent knowledge. The outer labels on the nodes (in italic) show the kinds of information that are involved in particular classes of decision on the medical domain.

It is quite easy to identify that there is a direct relationship between the knowledge organization (Figure 3) that we are suggesting and the nodes of the domino model. In fact, we can say that our architecture is an approximation of the domino model. In our case, the inference functions are determined by the answer sets of each frame.

5. CONCLUSIONS AND FUTURE WORK

In literature of cognitive research in medicine [8, 2], we can find suitable results which could be considered as guidelines in the design of intelligent systems in health care. In areas as declarative programming, we can find suitable approaches as *answer set programming* that permit to design and build intelligent systems whose behaviors can be close to the cognitive patterns that have been recognized by cognitive research in medicine.

As part of our research of building multi-agent systems in health care, in this paper we suggest a general architecture for building intelligent agents applied to health care. This architecture is based on an ontological model for clinical problem solving and answer set programming.

In our future work, we will define the technical details *w.r.t.* answer set semantics in order to build sound diagnostic-ASP Agents. Also as part of our future, we will explore the interaction diagnostic-ASP Agents in order to support decisions making. In fact, we are expecting to design intelligent agents able to diagnose if an organ is viable or not for transplanting [29].

Acknowledgements

We are grateful to anonymous referees for their useful comments. J.C. Nieves wants to thank CONACyT for his PhD Grant.

6. REFERENCES

[1] L. C. Aiello and F. Massacci. Verifying security protocols as planning in logic programming. *ACM Trans. Comput. Log.*, 2(4):542–580, 2001.

[2] J. F. Arocha, D. Wang, and V. L. Patel. Identifying reasoning strategies in medical decision making: A mehodological guide. *Journal of Biomedical Informatics*, 38:154–171, 2005.

[3] C. Baral. *Knowledge Representation, Reasoning and Declarative Problem Solving.* Cambridge University Press, Cambridge, 2003.

[4] C. Baral, K. Chancellor, N. Tran, N. Tran, A. M. Joy, and M. E. Berens. A knowledge based approach for representing and reasoning about signaling networks. In *ISMB/ECCB (Supplement of Bioinformatics)*, pages 15–22, 2004.

[5] G. Brewka, I. Niemelä, and T. Syrjänen. Implementing Ordered Disjunction Using Answer Set Solvers for Normal Programs. In *Proceedings of the 8th European Workshop Logic in Artificial Inteligence JELIA 2002.* Springer, 2002.

[6] S. DLV. Vienna University of Technology. http://www.dbai.tuwien.ac.at/proj/dlv/, 1996.

[7] T. Eiter, W. Faber, N. Leone, G. Pfeifer, and A. Polleres. A logic programming approach to knowledge-state planning, ii: The dlv^{k} system. *Artif. Intell.*, 144(1-2):157–211, 2003.

[8] D. A. Evans and C. S. Gadd. *Cognitive Science in Medicine*, chapter Managing Coherence and Context in Medical Problem-Solving Discourse, pages 211–255. The MIT Press, 1989.

[9] J. Fox and S. Das. *Safe and Sound: Artificial Intelligence in Hazardous Applications.* AAAI Press/ The MIT Press, 2000.

[10] M. Gelfond. *Handbook of Knowledge Representation*, chapter Answer Sets, pages 285–316. Elsevier, 2008.

[11] M. Gelfond and V. Lifschitz. The Stable Model Semantics for Logic Programming. In R. Kowalski and K. Bowen, editors, *5th Conference on Logic Programming*, pages 1070–1080. MIT Press, 1988.

[12] M. Gelfond and V. Lifschitz. Classical Negation in Logic Programs and Disjunctive Databases. *New Generation Computing*, 9:365–385, 1991.

[13] H. Jakobovits and D. Vermeir. Robust semantics for argumentation frameworks. *Journal of logic and computation*, 9(2):215–261, 1999.

[14] A. W. Kushniruk. Analysis of Complex Decision Process in Health Care: Cognitive Approaches to Health Informatics. *Journal of Biomedical Informatics*, 34:365–376, 2001.

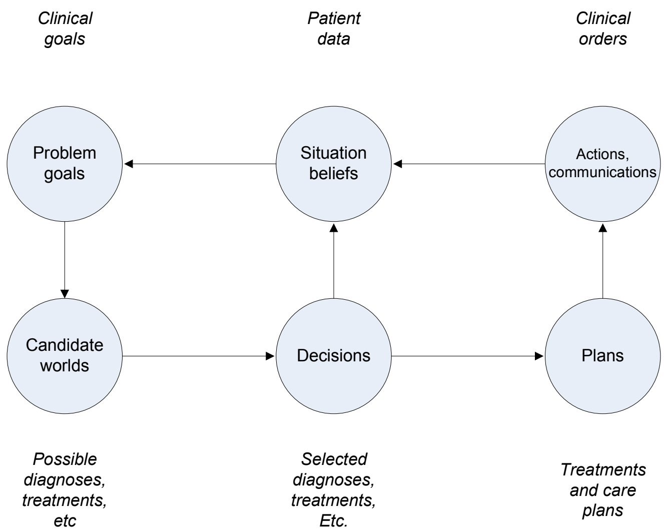

Figure 4: The domino agent model.

[15] P. Lucas. The Representation of Medical Reasoning Models in Resolution-based Theorem Provers. *Artificial Intelligence in Medicine*, 5(5):395–414, 1993.

[16] P. Nicolas, L. Garcia, and I. Stéphan. Possibilistic stable models. In *International Joint Conferences on Artificial Intelligence*, pages 248–253, 2005.

[17] J. C. Nieves, M. Osorio, and U. Cortés. Semantics for possibilistic disjunctive programs. In S. Costantini and R. Watson, editors, *Answer Set Programming: Advances in Theory and Implementation (ICLP-07 Workshop)*, pages 271–284, 2007.

[18] J. C. Nieves, M. Osorio, and U. Cortés. Semantics for possibilistic disjunctive programs (poster). In C. Baral, G. Brewka, and J. Schlipf, editors, *Ninth International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-07)*, number 4483 in LNAI, pages 315–320. Springer-Verlag, 2007.

[19] J. C. Nieves, M. Osorio, and U. Cortés. Preferred extensions as stable models. *Theory and Practice of Logic Programming*, accepted.

[20] J. C. Nieves, M. Osorio, U. Cortés, F. Caballero, and A. López-Navidad. Reasoning about actions under uncertainty: A possibilistic approach. In C. Angulo and L. Godo, editors, *In proceedings of CCIA*, 2007.

[21] M. Nogueira, M. Balduccini, M. Gelfond, R. Watson, and M. Barry. An a-prolog decision support system for the space shuttle. In *PADL*, volume 1990 of *Lecture Notes in Computer Science*, pages 169–183. Springer, 2001.

[22] M. Ortiz and M. Osorio. Nelson's strong negation, safe beliefs and the answer set semantics. In *Proc, of the third workshop on Answer Set Programming : Advances in theory and implementation*, 2005.

[23] M. Osorio, J. A. Navarro, and J. Arrazola. Applications of Intuitionistic Logic in Answer Set Programming. *Theory and Practice of Logic Programming (TPLP)*, 4(3):225–354, May 2004.

[24] M. Osorio, J. A. Navarro, J. R. Arrazola, and V. Borja. Ground nonmonotonic modal logic s5: New results. *Journal of Logic and Computation*, 15(5):787–813, 2005.

[25] D. Pearce. Stable Inference as Intuitionistic Validity. *Logic Programming*, 38:79–91, 1999.

[26] F. Ricca. The dlv java wrapper. In *APPIA-GULP-PRODE*, pages 263–274, 2003.

[27] S. SMODELS. Helsinki University of Technology. http://www.tcs.hut.fi/Software/smodels/, 1995.

[28] T. Syrjanen. A rule-based formal model for software configuration. reserach report A55, Helsinki University of Technology, 1999.

[29] P. Tolchinsky, U. Cortés, J. C. Nieves, A. López-Navidad, and F. Caballero. Using arguing agents to increase the human organ pool for transplantation. In *Proc. of the Third Workshop on Agents Applied in Health Care (IJCAI 2005)*, 2005.

[30] M. D. Vos and D. Vermeir. Extending answer sets for logic programming agents. *Annals of Mathematics and Artiffical Intelligence*, 42(1):103, 139 2004.

[31] C. Zepeda, M. Osorio, J. C. Nieves, C. Solnon, and D. Sol. Applications of preferences using answer set programming. In *Proc, of the third workshop on Answer Set Programming : Advances in theory and implementation*, Bath, UK, 2005.