ODDIN: Ontology-driven differential diagnosis based on logical inference and probabilistic refinements

Ángel García-Crespo [1], Alejandro Rodríguez [2], Myriam Mencke [2], Juan Miguel Gómez-Berbís [3], Ricardo Colomo-Palacios *

Department of Computer Science, Universidad Carlos III de Madrid, Av. Universidad 30, Leganes 28911, Madrid, Spain

ARTICLE INFO

Keywords:
Differential diagnosis
E-Science
Ontologies
Reasoning
Semantic Web

ABSTRACT

Medical differential diagnosis (ddx) is based on the estimation of multiple distinct parameters in order to determine the most probable diagnosis. Building an intelligent medical differential diagnosis system implies using a number of knowledge based technologies which avoid ambiguity, such as ontologies representing specific structured information, but also strategies such as computation of probabilities of various factors and logical inference, whose combination outperforms similar approaches. This paper presents ODDIN, an ontology driven medical diagnosis system which applies the aforementioned strategies. The architecture and proof of concept implementation is described, and results of the evaluation are discussed.

1. Introduction

The development of Medical Differential Diagnosis and Therapy systems using computational intelligence and distributed network technology has gained momentum over the last years (Zhao, Yanxiang, & Hui, 2005). In Cohen (2004), sciences, biology and medicine are considered to have been among the most progressive scientific fields during the 20th century, and such advancements are expected to have a tremendous impact on the information technology (IT) application domain landscape. However, leveraging the potential of knowledge intensive applications in medical differential diagnosis is a critical issue to be tackled in order to rely on the accuracy and efficiency of diagnosis or therapy systems.

Semantic Technologies (Berners Lee, Hendler, & Lassila, 2001) have emerged as an attempt to provide machine processable metadata to the ever increasing information resources on the Web. These software standards and methodologies may be applied to particular application domains in order to make maximum use of Semantic Web representation specifications such as RDF (W3C, 2004, 2006). Such specifications can define the terminology of a scientific domain as a computer interpretable ontology, using XML as the syntax for data interchange.

The semantic technologies which have been developed and improved alongside the advancement of the Semantic Web can be exploited to reveal machine readable latent relationships within specific diagnostical information in the medical discipline, where the homogeneity of terminology is particularly problematic (Fuentes Lorenzo, Morato, & Gómez, 2009). Ontologies provide a best of breed approach for addressing the aforementioned problem. The two pronged use of ontologies has the dual functions of allowing humans to grasp the meaning of any element having a well defined vocabulary and, secondly, having formal semantics to support reasoning. Using semantic technologies as the key technology enables data management of the vast amount of medical data (see, for example, García Sanchez, Fernandez Breis, Valencia García, Gómez, & Martínez Bejar, 2008; Gomez, Colomo Palacios, Mayoral, & Garcia Crespo, 2008).

This paper presents ODDIN, an ontology driven diagnosis system which applies differential diagnosis and probabilistic statistical refinements in order to solve the shortcomings of current ad hoc hardcoded systems, which do not benefit from the semantic possibilities of applications based on medical data.

The remainder of the paper is organized as follows. Section 2 outlines related research in the area, Section 3 describes the problem statement, detailing a typical scenario currently confronted by a user of diagnostical information and the problems which arise, thereby highlighting the application's relevance to the research presented. In Section 4, the architecture for the ODDIN approach is presented, which demonstrates technological assistance and potential solutions for problems in current medical diagnosis. The architecture has been implemented in the ODDIN prototype, the

* Corresponding author. Tel.: +34 91 624 5958; fax: +34 91 624 9129.
E-mail addresses: angel.garcia@uc3m.es (Á. García-Crespo), alejandro.rodriguez@uc3m.es (A. Rodríguez), myriam.mencke@uc3m.es (M. Mencke), juanmiguel.gomez@uc3m.es (J.M. Gómez-Berbís), ricardo.colomo@uc3m.es (R. Colomo-Palacios).
[1] Tel.: +34 91 624 9417.
[2] Tel.: +34 91 624 5936.
[3] Tel.: +34 91 624 5958.

proof of concept implementation for reasoning based diagnosis support. An evaluation using calculation of precision and recall rates of the system is also presented. Conclusions and future work are discussed in Section 5.

2. Related work

In the domain of medical diagnosis systems, a myriad of approaches exist, including various algorithmic techniques for automatic diagnosis that have been tested in research, as well as current systems presently available for use. Approaches in research which apply the use of combined techniques such as the current one include neuro fuzzy methods (Nauck & Kruse, 1999), the application of genetic algorithms (GAs) for rule selection (Ishibuchi, Nakashima, & Murata, 1999), or the unification of genetic algorithms with fuzzy clustering techniques (Roubos & Setnes, 2000). Other methods apply a single approach, applying neural networks, GAs, or fuzzy inference systems (e.g., Jang, 1993; Pena Reyes & Sipper, 1999; Song, Lee, Kim, & Park, 2005). A large number of medical diagnosis systems is also available, including DiagnosMD, DXPlain (London, 1998), eMedicine, Isabel (Ramnarayan et al., 2003), IWSMD (Ibrahim, AbdelRahman, & Farag, 2008), EasyDiagnosis, ADM (Le Beux et al., 1991), and Your Diagnosis. Other types of systems exist, such as those based on the PDP model (Saito & Nakano, 1988).

Regarding efforts which apply semantic techniques, initiatives such as OpenGalen (Rector, Rogers, Zanstra, & Van Der Haring, 2003) should be discussed, a not for profit organization which provides downloadable open source medical terminology. Other initiatives include for example OBO Foundries (Smith et al., 2007), a collaborative experiment among developers of science based ontologies. Within the scope of this research there are many resources in use such as Biological Ontologies (Lambrix, Tan, Jakoniene, & Strömbäck, 2007), Ontology based Support for Human Disease Study and Medical Ontologies to support human disease research and control (Hadzic & Chang, 2005), and Relations in biomedical ontologies (Smith et al., 2005).

Regarding rule based systems, a number of efforts have discussed automated extraction of rules as input to an expert system (Tsumoto, 1998). A large number of rule based systems are also applied to very different fields such as process controlling (Bernard, 1998), different process types optimizations (De Geus & Cohen, 1985) and operation plan creation (Bruno, Elia, & Laface, 1986), among others. Concerning the medical domain, different works have emerged which discuss treatment recommendation (Guyatt, Sinclair, Cook, & Glasziou, 1999), as well as those which analyze decision support systems in order to reduce clinical mistakes (Bates et al., 2001). Further efforts explain the benefits of the application of such systems (Hillestad et al., 2005; Kawamoto, Houlihan, Balas, & Lobach, 2005), from a theoretical perspective.

3. Motivating scenario

Differential diagnosis is a medical term which refers to a systematic method to identify a patient's illness based on determined signs and symptoms, by routinely assessing the symptoms and eliminating diagnoses until the most likely diagnosis which matches the symptoms is deduced. The principal objective of the research was to build a system which combines the strengths of probabilistic techniques, an ontology and logical inference, in order to aid a physician in the above process.

Another objective of the research was to construct the application as a tool for advanced students of medicine. In particular, a support system for Spanish students preparing for the Médico Interno Residente (MIR) examination. The MIR examination is a five hour exam drafted by the Spanish Department of Health and the Department of Education and Science, aimed towards students already holding 6 years university training in medicine[4] who wish to specialize. Those who pass the exam, which consists of 250 multiple choice questions, are subsequently permitted to practice their specialization as a specialist in the hospital where it is normally carried out. The MIR exam was initially introduced as a response to a surplus number of qualified medical graduates in Spain in the 1980s. A sample question for such a student may be formulated as follows.

A 46 year old male was admitted with intense chest pains and persistent elevation of the ST segment in sections II, III VF. Treatment for thrombosis was administered, and in the hours subsequent to treatment the symptoms evidenced were strain on the jugular vein, Kussmaul breathing, hepatomegalia, systolic artery tension of 70 nmiHg and normal pulmonary auscultation. What is the most likely diagnosis?

- Accelerated idioventricular rhythm
- Dressler syndrome
- Rupture of the papillary muscle
- Mayor ventricular aneurism
- Heart attack in the right ventricle

Examining the factors highlighted above, it is possible to formulate the question as a query. In this way, the system functions as a tool to solve the knowledge gap which emerges when a diagnosis is made exclusively based on subjective medical opinions, by performing an objective algorithmic analysis of all of the parameters. The method applies automated probabilistic statistical techniques and logical inference, which eliminates human subjectivity from the diagnosis process. The system enables analysis of every factor involved, assigning weights to each parameter introduced. The probability of each diagnosis is calculated as a function of the parameters. The inference engine consequently carries out logical inference, by consulting the ontology of diseases and associated signs and symptoms. The inference process, combined with the probabilities, generates a ranked list of most likely diagnoses. Despite the existence of similar systems (these are discussed in Section 2), the current system performs analyses for scenarios such as the one above, using a new combination of techniques not previously available to students of medicine or doctors.

4. ODDIN

Given the aforementioned problems typical of traditional systems, the current section describes the approach. It is based on several design principles to solve the drawbacks of previous disintegrated applications, and built as a medical diagnosis system. The principal architecture of the medical diagnosis application was initially constructed and named Ontology DDx (ODDX). However, the research has been entitled ODDIN, in order to highlight the novel combination of an ontology and logical inference for differential diagnosis. The main objective of the architecture and its structure will be described in the sections which follow.

4.1. ODDX

ODDX is a clinical diagnostics software. The objective of this tool is to provide the user with an expert system which enables

[4] The approximately equivalent exams in the United States are the United States Medical Licensing Examinations step 1, United States Medical Licensing Examinations step 2, followed by the Clinical Skills Assessment, and a Test of English as a Foreign Language (TEFL).

the determination of different medical diagnostics generated from a collection of indications.

The diagnostics produced by the system cannot be the only factor used to come to a decision, and must be accompanied by the knowledge of an expert medical user. However, this diagnosis tool can provide invaluable assistance when making medical decisions. The tool is realized in the form of a medical support system, which suggests a set of diagnoses to the user, ranked in terms of more or less probable. The probabilities are defined in terms of percentages which are based on indications introduced to the system.

4.2. System architecture

The structure of ODDX is not based on a traditional architecture, such as a client server system. The idea of the application provided by ODDX was to build all of the components which perform the diagnosis process as a group of modules or engines, which when integrated allow their combination to infer new knowledge and realize the diagnosis. Fig. 1 displays the architecture of ODDX.

Fig. 1 graphically demonstrates each component of the architecture, and displays its position in the system. Each constituent will now be described in further detail below.

4.2.1. System architecture constituents

The left hand side of the diagram shows the module called "ODDX", which can be considered the main application. Behind the main module, the different engines or systems which ODDX uses for the correct working of application can be seen. Each component is explained in detail below, entitled according to its function.

4.2.2. Probabilistic system

The probabilistic component is the system that is responsible for managing and/or calculating the probabilities of every diagnostic inferred. Every disease that is diagnosed (one or more) from a group of indications has its own probability of being true or not. This probability, and a detailed breakdown of it (individual probabilities are assigned to indications, which result in a diagnosis), will be calculated by the probabilistic system. The functioning of the probabilistic system will be explained further in Section 4.4.

4.2.3. Data loading

This is the engine which performs data loading from the ontology. It employs the Jena API[5] to read the ontology file, which is an open source Java programming environment for Semantic Web applications, and supports the use of various languages, such as Resource Description Framework (RDF), Resource Description Framework Schema (RDFS), Ontology Web Language (OWL) and Query Language For RDF (SPARQL). The ontology file contains all of the data needed to make the diagnosis, such as diseases, symptoms, and drugs. These data will be loaded to memory so that the user can interact easily with the application. The ontology was built subsequent to having consulted numerous reputable sources of medical information (Ausina Ruiz & Moreno Guillén, 2006; Hoeprich, Jordan, & Ronald, 1994; Kasper et al., 2004), and having carried out interviews with medical professionals and students.

4.2.4. Combination system

The combination system computes all of the diagnostic combinations possible which may be the result of the interaction of drugs. Basically it is a method which allows, given a patient with a group of indications and associated drugs, the calculation of the possible interactions which may be caused by drugs.

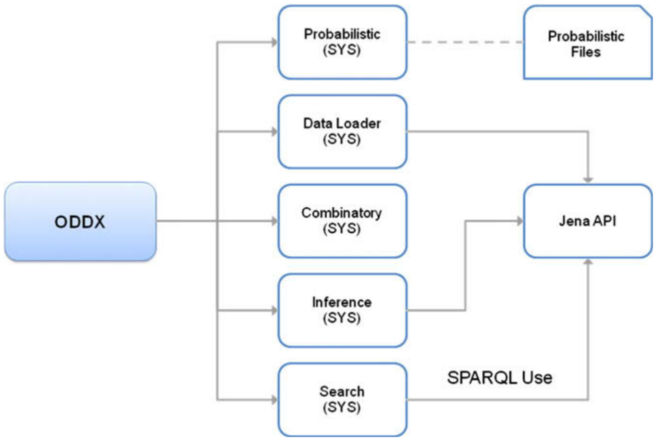

Fig. 1. ODDX system architecture.

4.2.5. Inference

The inference engine is the main engine of application, because it is principal constituent of the system which really enables the diagnoses to be made. This engine requires access to the knowledge base with the diseases, symptoms, etc., and at the same time needs access to the knowledge base that contains inference rules. With the conjunction of these items and the Jena API, ODDX performs the inferences which result in the diagnosis. Thus, the system behaves as a rule based expert system. It is able to acquire new knowledge and increase the reasoning speed using techniques such as incremental reasoning (Parsia, Halaschek Wiener, & Sirin, 2006).

4.2.6. Search

This component takes the form of a search engine, which realizes SPARQL queries to the ontology to consult all of the data stored it. This permits fast access to all of the data stored within the ontology.

In the diagram, it is also possible to view how some of the engines make use of the Jena API, which also consults the ontology file. This is because these engines need to use this API in order to realize the actions associated with them. It can also be seen that the probabilistic engine makes use of a probabilistic file collection. This is because individual probabilities associated to every disease are stored in a group of files, which contain the probabilities.

4.3. Reasoning and inference

The effective performance of ODDX relies on the reasoning and inference capabilities offered by ontologies. The principal constituents of its functioning may be divided into two elements:

- An ontology based on differential diagnostics.
- Inference rules, which enable reasoning to derive the correct knowledge.

Specifically, the components above have been built using Jena Rules. For the current research, these rules have been applied, however, in future applications, it may also be possible to export rules to alternative formats, such as SWRL. Initially, the global functioning of ODDX will be described. This will be followed by an explanation of the behaviour of the inference rules and thus reasoning used. Fig. 2 displays the main working components of ODDX, showing how it functions globally. Subsequent to the diagram, the rules employed are described.

[5] JENA Semantic Web Framework. http://jena.sourceforge.net/index.html. Last Verified: 26th December, 2008.

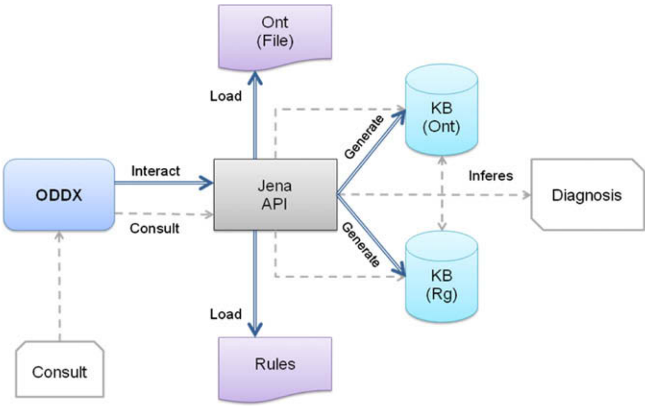

Fig. 2. System overview.

4.3.1. *System knowledge base architecture*

The upper half of Fig. 2 demonstrates the generation of the knowledge bases. Firstly, ODDX interacts with the Jena API, which at the same time loads two files: a rules file and an ontology file, and it generates two knowledge bases, the ontology knowledge base and the rules knowledge base. These two items generated are necessary for knowledge inference, and are the basic pillars underlying ODDX́s behaviour as an expert system.

A working example of the inference system of ODDX can be seen in the lower half of Fig. 2. When a query is realized, ODDX manages it. This management is performed by consulting the Jena API, which interacts with the two knowledge bases generated previously and infers the reasoning which the knowledge bases provide.

4.3.2. *Rules*

The fundamental component of the application will now be described, that is, the inference system which provides the rules. In some cases, this part of the system will be compared with other reasoning possibilities; such as logical descriptions (Grosof, Horrocks, & Decker, 2003). Inference rules allow the establishment of the same parameters as logical descriptions rapidly, however, they also offer some advantages suitable for the current application.

With the application of Jena rules, similarly to the process for logical descriptions, firstly a set of rules for every disease which can be inferred by the system must be established. However, contrary to logical descriptions, the rules do not need to be stored in the Web Ontology Language (OWL) ontology file. These rules can be stored in another file and can be loaded separately. A number of examples of the functioning of the rules are detailed below.

Specifically, the number of rules that must be generated are:

$$Rules\_number \quad n + m + 2$$

where n is the number of symptoms, and m is the number of laboratory tests.

The rules have the following format:

@prefix ont: <URI ONTOLOGIA#>.
@include <RDFS>.[rule DIS P NOT REST SYMPTOMS:
(?i ont:hasSymptom ?x) notEqual(?x, ont:SYM A) notEqual(?x, ont:SYM E)
< (?i ont:hasNegSymptom ont:DIS DIS P NOT SYM)]

In the above case, rules are defined which establish premise that must define the ''hasNegSymptom'' property for ''not valid'' symptoms. For example, SYM_A or SYM_E and the only valid symptoms, so, the rule must be applied in the rest of symptoms. If the rule receives some other ''not valid'' symptom (Any other symptom except SYM_A or SYM_E), it will establish the hasNegSymptom property.

This procedure may essentially be defined as a symptom discriminate.

[rule DIS P NOT REST LABTESTS:
(?i ont:hasLabTest ?x) notEqual(?x, ont:LT 1)
> (?i ont:hasNegLabTest ont:DIS_DIS_P_NOT_LABT)]

In the case of the rule above, it performs in the same way as the previous rule, the difference being that in this case the laboratory tests that not are defined for the disease are ''discriminated''.

[rule DIS DIS P SYM A:
(?i ont:diagnosis ont:DIS P)
< (?i ont:hasSymptom ont:SYM A)
noValue(?i, ont:hasNegSymptom ont:DIS DIS P NOT SYM) noValue(?i, ont:hasNegLabTest ont:DIS DIS P NOT LABT)]

The next rule (backward form rule), shown above, allows to infer the disease being dealt with (DIS_P), in the case that it is specified that a valid symptom (SYM_A) is being inferred and not a ''not allowed'' one (this rule assigns ''not allowed'' to symptoms and laboratory tests).

[rule DIS DIS P SYM E:
(?i ont:diagnosis ont:DIS P)
< (?i ont:hasSymptom ont:SYM E)
noValue(?i, ont:hasNegSymptom ont:DIS DIS P NOT SYM) noValue(?i, ont:hasNegLabTest ont:DIS DIS P NOT LABT)]

The rule above shows the same reasoning process as the previous rule, but for the SYM_E symptom.

[rule DIS DIS P LT 1:
(?i ont:diagnosis ont:DIS P)
< (?i ont:hasLabTest ont:LT 1)
noValue(?i, ont:hasNegSymptom ont:DIS DIS P NOT SYM) noValue(?i, ont:hasNegLabTest ont:DIS DIS P NOT LABT)]

The final rule also functions in the same way as the previous rule, but in this case it is applied to the LT_1 laboratory test.

It can be observed from the examples of the rules described above that a number of types of rules are required. A rule must be defined for each symptom and each laboratory test, as well as rules specifying the symptoms and laboratory tests which must not be present, which in this way ensure their exclusion. The rules may be classified into two types: rules in backward form and rules in forward form.

No additional calculations or computational complexities are required for the application of the rules. This is in contrast to description logic, where a cardinality restriction must be established to make the Closed World Assumption (CWA). By making use of rules, defining such restrictions is not necessary, because the CWA is already present in the rules, by the application of the rules which exclude those symptoms and laboratory tests that are not necessary.

4.4. *Probabilistic computation*

The internal working of the probabilistic engine is based on Bayes Theorem (Smets, 1987), with some modifications in order to adapt it to the computational requirements of the system. The adaptation of this theorem was made using (Martin & Del Castillo, 2004). To perform the probabilistic analysis, a weight for every diagnosis variable needs to be established. This weight indicates the importance of every variable in the final diagnosis. In the current case, the following variables are considered: Symptoms/Signs, Laboratory tests, Sex, Age, Countries visited, Transfusions, and Operations. In this case medicines are not introduced as variables, as the interaction of medicine is processed differently within the system using logical inference (this will be described in further detail below). All of the variables must have a weight between 0 and 1, and similar to the model proposed by Bayes theorem and other probabilistic models, the total of the weights when summed must equal 1. Below an example of the functioning of the probabilistic engine is provided, which assumes the following weights:

- Symptoms: 0.3 30%
- Labtests: 0.5 50%
- Sex: 0.035 3.5%
- Age: 0.035 3.5%
- Countries visited: 0.1 10%
- Transfusions: 0.015 1.5%
- Operations: 0.015 1.5%

In the current implementation of the system, these values are stored in a config file which enables the user to obtain values by means of a number of simple probabilistic computations. An example of the representation of the weights is demonstrated below.

```
################################
# Oddx: Probabilistic System #
################################
SYMPTOMS    0.3
LABTESTS    0.5
SEX    0.035
AGE    0.035
COUNTRIES    0.1
TRANSFUSIONS    0.015
OPERATIONS    0.015
ADD_PROBABILITY_BY_DEFAULT    true
```

In what follows, it is explained how the engine works, by demonstrating how it treats the variables. Regarding the processing of symptoms, the system infers diseases from symptoms and labtests. Suppose a case where the following symptoms and weights are generated:

- Sym A   Weight: Medium
- Sym B   Weight: High
- Sym C   Weight: Very high

Then assume that the system infers diseases "X" and "Y" based on these values. It should be noted that the system infers the diseases as a function of their symptoms, using the ontology within the system. Weights are used to establish probability.

To calculate the probability of disease "X", the following computations are performed. All of the symptoms which caused the system to infer disease "X" are considered. The symptoms are "A", "B", and "C". The probability of each of these symptoms in disease X must be established. It is assumed that every symptom will be processed separately, therefore, the following values must be obtained:

- Probability of "A Symptom" with "Medium" weight on "Disease X".
- Probability of "B Symptom" with "High" weight on "Disease X".
- Probability of "C Symptom" with "Very High" weight on "Disease X".

The reason for this calculation is because for example the probability of "High fever" in "Flu" is not the same that "High Fever" in "Cholera", thus a distinction between symptoms and diseases must be made. The solution provided to implement this in the system is to read a configuration file which connects every disease with its symptoms and associated weights. To model this procedure, the following system is created. Every disease has a folder marked by a code which indicates that particular disease. Suppose that in the previous example, the code for disease "X" is "CDX". So, there will be a folder in the system entitled "CDX". Inside the folder of this disease there will be a large number of files with symptoms and labtests associated with this disease, each entitled with a code for each of the symptoms and labtests. Continuing the example, disease "X" was inferred because it has symptoms "A", "B" and "C", thus the files "CDA.prob", "CDB.prob" and "CDC.prob" will be generated inside the folder "CDX".

The final outcome is as follows. The probability of the symptom, in the particular disease, with an associated concrete weight, will be generated in every file. The file is structured as in the example below:

```
VERY_LOW = 0.63
LOW = 0.25
MEDIUM = 0.32
HIGH = 0.175
VERY_HIGH = 0.115
```

Therefore, if for example, this is the content of "CDA.prob" file, it can be determined that "The probability that disease 'X' has the symptom 'A' with weight 'Medium' is 0.32". In this case it can be observed that the addition of all the probabilities does not need to sum to 1. This is because in this case only one option is selected. Only the probability of a custom symptom with a custom weight in a custom disease is established. The other options are not used. Furthermore, the addition of probabilities will be divided between the number of symptoms, so the maximum possible value will be 1, and must be multiplied (as can be seen in the next lines) by the weight that represents the symptoms, so the maximum possible probability will be the maximum possible for the symptoms. Regarding the rest of the symptoms, the same results will be obtained. Suppose now that the following results for disease X with symptoms "A", "B" and "C" with the aforementioned weights are obtained:

Disease X:

A [Weight: Medium] → Prob: 0.32
B [Weight: High] → Prob: 0.57
C [Weight: Very high] → Prob: 0.98

If the probabilities are summed, and divided by the number of symptoms:

$(0.32 + 0.57 + 0.98/3) \quad 0.6233333$

This is the percentage of the symptoms in this disease, but it must be considered that symptoms have a certain weight with respect to the final result. Returning to the previously described overview of the probabilistic system, the following line can be observed:

SYMPTOMS   0.3

The symptoms represent a 30% of total probability in diagnosis. Therefore, the following mathematical operation needs to be performed:

$$0.6233333 * 0.3 \quad 0.18699999 \quad 18.69\%$$

The final outcome of the operation is that in disease ‘‘X”, symptoms represent 18.69% of the final diagnostic. The functioning is exactly the same for the laboratory tests parameter, but this factor only has two weights: Positive and Negative. Regarding the processing of the ‘‘countries visited” parameter, it is similar to the process described above, with minor modifications. In this case, what is required is the lookup of the probability of a custom disease in a certain country (for example: P1 and P2). This is because the probability of, for example, Cholera is not the same in ‘‘Spain” as in ‘‘Nigeria”. So, the probability that disease X exists in country ‘‘Pi” needs to be established. These probabilities will be added up, and the result will be divided between the number of countries to be finally multiplied by the weight assigned to the countries. The implementation is similar to that of the symptoms. When the disease is established, the code is also known, and thus the folder ‘‘CDX” is accessed. However, in the current case, the file ‘‘countries.prob”, is accessed, which will have a format as follows:

```
AFG = 0.1239
ALA = 0.5534
ALB = 0.7953
ANT = 0.6508
...
etc.
```

This file contains the codes of all the countries of the world, and the probability that the disease being consulted exists in the country under consideration.

With regard to other variables, the sex, age, transfusions and operations only admit one value: ‘‘true or false”. Age, for example, can be ‘‘within the rank or not”. In the case of age, it is verified whether or not the disease that the system infers permits the age introduced by the user inside its rank, and if so, the probability is 1, that is, 100% for this variable. However, to avoid generating a result higher than 100%, the result 1 determined previously is multiplied by the weight associated with age, which in the outline of the system was 0.035. Thus, the final outcome of the weight of age will be 0.035 (3.5%). If age case is not fulfilled, that is, the age input by the user was not the same as the age range of the disease, a 0 will be assigned, and when multiplied by its weight, the final probability would be 0. The same process will be repeated with all the variables, and the results will be summed to obtain the final probability.

There is one final parameter in the system. In the configuration file of the probabilistic engine a parameter called ‘‘ADD_PROBABILITY_BY_DEFAULT” can be seen. When this parameter is set to ‘‘true” value, the system adds a default probability if a particular parameter is not found. If for example the ‘‘sex” value is not introduced in a query, or a predefined value for sex does not exist (indifferent), if this option is enabled, the system will be add a default probability: 0.035. If this option is disabled, it will add nothing.

4.5. Evaluation

The current section provides a detailed description of the parts of the ODDIN system which should be evaluated in order to accurately analyze it, as well as the results obtained. A comprehensive analysis is detailed of the results acquired, based on a sample of distinct users who evaluated the system. An evaluation of the usability of the system was performed, based on a series of questionnaires administered to the test users.

4.5.1. Evaluation of diagnosis results

In order to realize the evaluation of the output of the system, two techniques were applied. On the one hand, it was aimed to verify whether the system correctly inferred results as a function of the inference rules used. This evaluation was necessary to determine whether the system functioned correctly with the rules given, independently of the accuracy of the rules.

In order to correctly realize the evaluation, determined test instances were created for which in various cases the system should arrive at the conclusion of one of these instances as a function of the test cases which it receives as input. These instances are displayed in Fig. 3. The criteria for the inclusion of the instances in order to perform the tests were that the entire set of instances covered all of the possible diagnoses which could be generated at the moment of determining a given diagnosis. These possible diagnoses included a possible empty diagnosis (insufficient data available to conclude the diagnosis), or a diagnosis where various results may be obtained, among other possibilities.

The content validity of a construct is the extent to which the set of measures provides adequate coverage for the construct domain or essence of the domain being measured (Churchill, 1979). Some authors (Emory, 1985) pointed out that the determination of content validity is not numerical, but subjective and judgmental. Taking this into account, as part of the current study feedback from three academics which have a deep understanding of higher education issues was requested. All of them were asked about the clarity of the questions and measurement items used to evaluate the current system. As a result of this process, some items where rewritten to adapt questionnaire content to experts’ opinions.

To carry out the evaluation of the system, the experts selected to use the system were general practitioners (GPs), given that the system is a general support system which does not apply to any specialized medical field.

The instances in Fig. 3 constitute the domain of testing used to evaluate the performance of the inference of the system. The concrete test cases are those shown in Fig. 4.

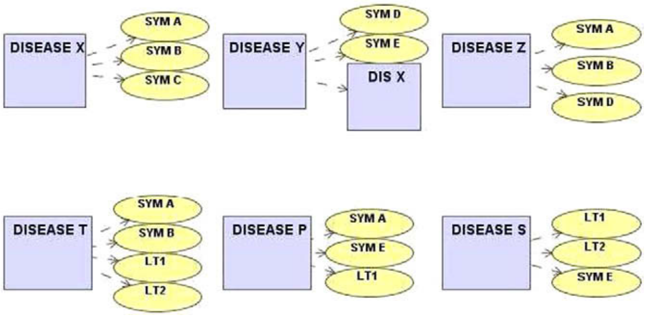

Fig. 3. Test individuals.

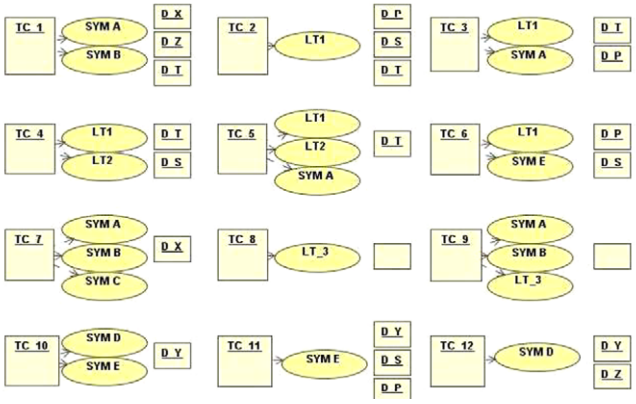

Fig. 4. Test cases.

In this example, the test cases are represented by "TC", and their associated symptoms (SYM), laboratory tests (LT), as well as the results which the system should generate. The tests indicated that in 100% of cases, the results inferred were correct.

It was possible to further verify whether the results were correct by consulting the inference rules established for each illness. These rules were established as a function of some basic criteria: Collection of symptoms → Illness(es). Adding a new symptom to the previous set generated the consequence that some of the illnesses diagnosed disappeared, given the impact of the new symptom on the diagnosis.

N° experts: 28

N° tests/expert: 4(3 + 1 Usability)

To evaluate the performance of annotation, standard recall, precision and F1 measures were used. Recall and precision measures reflect the different aspects of annotation performance. These measures were first used to measure an Information retrieval system by Cleverdon, Mills, and Keen (1966). F1 measure was later introduced by van Rijsbergen (1979) in order to combine precision and recall measures, with equal importance, into a single parameter for optimization. Precision, Recall and F1 measures are defined as follows:

$$Precision \quad \frac{Diseases\_found\_and\_correct}{Total\_Found}$$

$$Recall \quad \frac{Diseases\_found\_and\_correct}{Total\_Correct}$$

$$F1 \quad \frac{2 * Precision * Recall}{Precision + Recall}$$

The following data were obtained:

Diseases found and correct: 287
Total found: 331
Total correct: 336

The values obtained for precision, recall and F1 are the following:

Precision = 0.867
Recall = 0.854
F1 = 0.86

The results above were calculated based on Tables 1 3, which represent the tests performed by the test users (Figs. 5 7).

Table 1
Results of the tests.

Values	P1	P2	P3
All correct results	26	20	3
Missing results	0	0	21
Extra results	2	8	0
Missing and extra results	0	0	4

Table 2
Usability results.

Values	Usability
Very bad	1
Bad	5
Average	4
Good	10
Very good	8

Regarding the results of the evaluation with respect to precision and recall, it should be considered that in the application environment of the system the values achieved should be optimal (in particular, recall). That is, the values should be 1, or as close to 1 as possible, given that even though these systems are decision support systems, their correct functioning is critical.

Table 3
ODDIN statistical results.

Data	Average	Std. deviation
P1	1.14	0.524
P2	1.57	0.920
P3	2.17	0.818
Usability	2.78	0.875

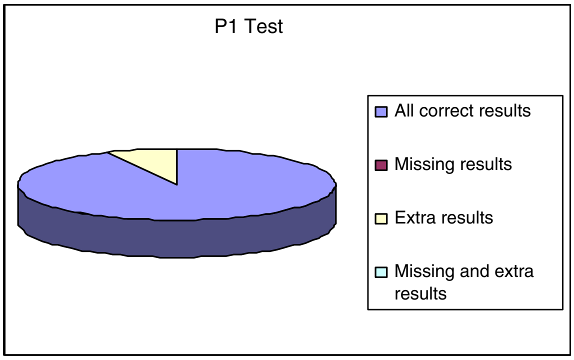

Fig. 5. Results test 1.

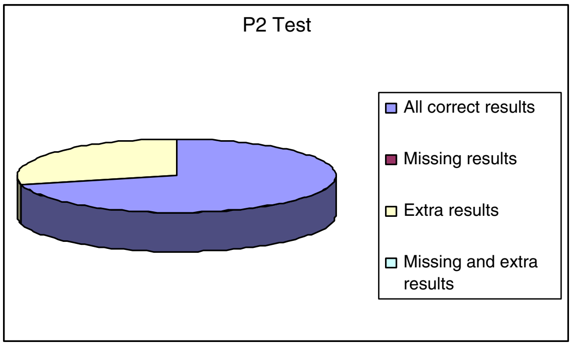

Fig. 6. Results test 1.

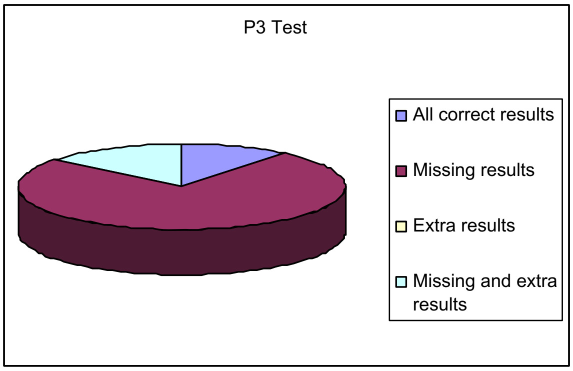

Fig. 7. Results test 1.

5. Conclusions and future research

This paper has described a system for differential diagnosis in medicine using ontologies, which supports physicians and students of medicine. After evaluation, it can be concluded that the system holds a number of characteristics which differentiates it from other similar applications, including ease of use, the application of medical standards through rigorous consultation of literature, more clinical parameters, and the integration of an ontology and a probabilistic system.

Initial performance expectations of the system were fulfilled. For future versions of the system, it is intended to host it as a Web application and build a more extended ontology, as well as amplifying the databases which store the probabilistic parameters. Future research is focused on integrating a component which investigates the effect of drugs on the diagnosis, and take account of such secondary effects by including them as probabilistic parameters. This functionality is currently being built into the system as ongoing research.

Acknowledgements

This work is supported by the Spanish Ministry of Industry, Tourism, and Commerce under the project SONAR (TSI 340000 2007 212), GODO2 (TSI 020100 2008 564) and SONAR2 (TSI 020100 2008 665), under the PIBES project of the Spanish Committee of Education & Science (TEC2006 12365 C02 01) and the MID CBR project of the Spanish Committee of Education & Science (TIN2006 15140 C03 02).

References

Ausina Ruiz, V., & Moreno Guillén, S. (2006). *Tratado SEIMC de Enfermedades Infecciosas y Microbiología Clínica (SEIMC treatise of Infectious Diseases and clinic microbiology)*. Buenos Aires: Médica Panamericana.

Bates, D. W., Cohen, M., Leape, L. L., Overhage, J. M., Shabot, M. M., & Sheridan, T. (2001). Reducing the frequency of errors in medicine using information technology. *Journal of American Medical Informatics Association, 8*(4), 299–308.

Bernard, J. A. (1998). Use of a rule-based system for process control. *IEEE Control Systems, 8*(5), 3–13.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American, 284*(5), 34–44.

Bruno, G., Elia, A., & Laface, P. (1986). A rule-based system to schedule production. *Computer, 19*(7), 32–40.

Cohen, J. (2004). Bioinformatics – An introduction for computer scientists. *ACM Computing Surveys, 36*(2), 122–158.

Churchill, G. A. Jr., (1979). A paradigm for developing better measures of marketing constructs. *Journal of Marketing Research, 16*, 64–73.

Cleverdon, C. W., Mills, J., & Keen, E. M. (1966). *Factors determining the performance of indexing systems*. Cranfield, UK: College of Aeronautics.

De Geus, A. J., & Cohen, W. (1985). A rule-based system for optimizing combinational logic. *IEEE Design & Test, 2*(4), 22–32.

Emory, W. C. (1985). *Business research methods* (3rd ed.). Homewood, IL: Irwin.

Fuentes-Lorenzo, D., Morato, J., & Gómez, J. M. (2009). Knowledge management in biomedical libraries: A Semantic Web approach. *Information Systems Frontiers, 11*(4), 471–480.

García Sanchez, F., Fernandez Breis, J., Valencia García, R., Gómez, J. M., & Martínez-Bejar, R. (2008). Combining Semantic Web technologies with multi-agent systems for integrated access to biological resources. *Journal of Biomedical Informatics, 41*(5), 848–859.

Gomez, J. M., Colomo-Palacios, R., Mayoral, M. R., & Garcia-Crespo, A. (2008). Micro-array information and data integration using SAMIDI. In J. R. Rabuñal, J. Dorado, & A. Pazos (Eds.), *Encyclopedia of artificial intelligence*. Hershey, PA: IGI Global.

Grosof, B. N., Horrocks, I., & Decker, S. (2003). Description logic programs: Combining logic programs with description logic. In *12th international conference on the World Wide Web (WWW-2003), Budapest, Hungary* (pp. 48–57).

Guyatt, G. H., Sinclair, J., Cook, D. J., & Glasziou, P. (1999). Users' guides to the medical literature XVI. How to use a treatment recommendation. *Journal of the American Medical Association, 281*(19), 1836–1843.

Hadzic, M., & Chang, E. (2005). Medical Ontologies to support human disease research and control. *International Journal of Web and Grid Services, 1*(2), 139–150.

Hillestad, R., Bigelow, J., Bower, A., Girosi, F., Meili, R., Scoville, R., et al. (2005). Can electronic medical record systems transform health care? Potential health benefits, savings, and costs. *Economics of Health Information Technology, Health Affairs, 24*(5), 1103–1117.

Hoeprich, P. D., Jordan, M. C., & Ronald, A. R. (1994). *Infectious diseases: A treatise of infectious processes*. Philadelphia: J.B. Lippincott Co.

Ibrahim, M., AbdelRahman, S., & Farag, I. (2008). IWSMD: An Intelligent Web Service based Medical Diagnosis System. In *The 6th international conference on informatics and systems 27–29 March 2008, El Cairo, Egypt.*

Ishibuchi, H., Nakashima, T., & Murata, T. (1999). Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B, 29*(5), 601–618.

Jang, J. (1993). ANFIS: Adaptive-network based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics, 23*(3), 665–685.

Kasper, D. L., Braunwald, E., Fauci, A., Hauser, S., Longo, D., & Jameson, J. L. (2004). *Harrison's principles of internal medicine* (16th ed.). New York: McGraw-Hill.

Kawamoto, K., Houlihan, C. A., Balas, E. A., & Lobach, D. F. (2005). Improving clinical practice using clinical decision support systems: A systematic review of trials to identify features critical to success. *British Medical Journal, 330*, 765–768.

Lambrix, P., Tan, H., Jakoniene, V., & Strömbäck, L. (2007). Biological ontologies. In C. J. O. Baker & K. H. S. Cheung (Eds.), *Semantic Web revolutionizing knowledge discovery in the life sciences* (pp. 85–89). Berlin: Springer.

Le Beux, P., Rammal, M., Riou, C., Frangeul, C., Cador, F., & Lenoir, P. (1991). ADM: A diagnostic aid knowledge base for general practitioners. In *IMIA international conference on medical informatics and medical education* (pp. 223–229).

London, S. (1998). DXplain: A Web-based diagnostic decision support system for medical students. *Medical Reference Services Quarterly, 17*(2), 17–28.

Nauck, D., & Kruse, R. (1999). Obtaining Interpretable Fuzzy Classification Rules from Medical Data. *Artificial Intelligence in Medicine, 16*(2), 149–169.

Martin, A., & Del Castillo, J. D. (2004). *Bioestadística para las Ciencias de la Salud (Science health biostatistics)*. Madrid: Ediciones Norma.

Parsia, B., Halaschek-Wiener, C., & Sirin, E. (2006). Towards incremental reasoning through updates in OWL-DL. In: *Proceedings of the workshop at 15th international World Wide Web conference on reasoning on the Web.*

Pena-Reyes, C., & Sipper, M. (1999). Designing breast cancer diagnostic system via a hybrid fuzzy-genetic methodology. In *Proceedings of IEEE international fuzzy systems conference* (pp. 135–139).

Ramnarayan, P., Tomlinson, A., Rao, A., Coren, M., Winrow, A., & Britto, J. (2003). ISABEL: A web-based differential diagnostic aid for paediatrics: Results from an initial performance evaluation. *Archives of Disease in Childhood, 88*(5), 408–413.

Rector, A. L., Rogers, J. E., Zanstra, P. E., & Van Der Haring, E. (2003). OpenGALEN: Open source medical terminology and tools. In *Proceedings of the American medical informatics association symposium 2003* (pp. 982–985).

Roubos, J. A., & Setnes, M. (2000). Compact Fuzzy Models through Complexity Reduction and Evolutionary Optimization. In *Proceedings of IEEE international conference on fuzzy systems* (pp. 762–767). USA: San Antonio.

Saito, K., & Nakano, R. (1988). Medical diagnostic expert system based on PDP model. In *IEEE international conference on neuronal networks* (pp. 255–262).

Smets, P. (1987). Belief functions: The disjunctive rule of combination and the generalized Bayesian theorem. *International Journal of Approximate Reasoning, 9*(1), 1–35.

Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., et al. (2007). The OBO Foundry: Coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology, 25*(11), 1251–1255.

Smith, B., Ceusters, W., Klagges, B., Köhler, J., Kumar, A., Lomax, J., et al. (2005). Relations in biomedical ontologies. *Genome Biology, 6*(5), 1425–1433.

Song, H., Lee, S., Kim, D., & Park, G. (2005). New methodology of computer aided diagnostic system on breast cancer. In *Proceedings of international symposium on neural networks* (pp. 780–789).

Tsumoto, S. (1998). Automated extraction of medical expert system rules from clinical databases based on rough set theory. *Information Sciences: An International Journal, 112*(1–4), 67–84.

Van Rijsbergen, C. J. (1979). *Information retrieval*. Newton, MA: Butterworth-Heinemann.

W3C. (2004). *RDF Vocabulary Description Language 1.0: RDF Schema* <http://www.w3.org/TR/rdf-schema/>.

W3C. (2006). *Resource description framework (RDF)* <http://www.w3.org/RDF/>.

Zhao, W., Yanxiang, H., & Hui, J. (2005). A model of intelligent distributed diagnosis and therapy system based on mobile agent and ontology. In *Proceedings of the 8th international conference on high-performance computing in the Asia-Pacific Region*.