

Decisional DNA: A multi-technology shareable knowledge structure for decisional experience

Cesar Sanin^{a,*}, Carlos Toro^b, Zhang Haoxi^a, Eider Sanchez^b, Edward Szczerbicki^a, Eduardo Carrasco^b, Wang Peng^a, Leonardo Mancilla-Amaya^a

^a Faculty of Engineering and Built Environment, The University of Newcastle, University Drive, Callaghan, NSW, Australia

^b Vicomtech-IK4, San Sebastian, Spain

ARTICLE INFO

Keywords:

Decisional DNA
Set of experience knowledge structure
Knowledge representation
Knowledge engineering
Decision making
Artificial intelligence

ABSTRACT

Knowledge representation and engineering techniques are becoming useful and popular components of hybrid integrated systems used to solve complicated practical problems in different disciplines. These techniques offer features such as: learning from experience, handling noisy and incomplete data, helping with decision making, and predicting capabilities. In this paper, we present a multi-domain knowledge representation structure called Decisional DNA that can be implemented and shared for the exploitation of embedded knowledge in multiple technologies. Decisional DNA, as a knowledge representation structure, offers great possibilities on gathering explicit knowledge of formal decision events as well as a tool for decision making processes. Its applicability is shown in this paper when applied to different decisional technologies. The main advantages of using the Decisional DNA rely on: (i) versatility and dynamicity of the knowledge structure, (ii) storage of day-to-day explicit experience in a single structure, (iii) transportability and shareability of the knowledge, and (iv) predicting capabilities based on the collected experience. Thus, after analysis and results, we conclude that the Decisional DNA, as a unique multi-domain structure, can be applied and shared among multiple technologies while enhancing them with predicting capabilities and facilitating knowledge engineering processes inside decision making systems.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Now-a-days, engineering systems are established on evolving paradigms; knowledge and users' experience take a big role in today's applications as we have now the computational potential of modeling such paradigms. The term knowledge engineering (KE) has been defined as a discipline that aims to offering solutions for complex problems by the means of integrating knowledge into computer systems [1]. It involves the use and application of several computer science domains such as artificial intelligence, knowledge representation, databases, and decision support systems, among others. Knowledge engineering technologies make use of the synergism of hybrid systems to produce better, powerful, more efficient and effective computer systems. Among the features associated with knowledge engineering systems are human intelligence capabilities such as learning, reasoning and forecasting from current knowledge or experience. From an application point of view, different research projects

have been presented by the scientific community involving knowledge representation and decision making technologies to extend the user's understanding; however, to our acquaintance, most of these approaches miss the potential of using knowledge based theories that might enhance the user's experience and at the same time creating his/her decisional fingerprints.

In our case, we propose experience as the main and most appropriate source of knowledge and its use leads to useful systems with improved performance. Multiple applications perform decisions, and most of the decisions are taken in a structured and formal way, this is what we call formal decision events. All these formal decision events are usually disregarded once the decision is made, or even worse, if the system is queried again, the decision has to be repeated. What to do with the experience gained on taking such decisions relies on our proposed knowledge representation structure. We propose the Decisional DNA as a unique and single structure for capturing, storing, improving and reusing decisional experience. Besides, we make use of the Set of Experience (SOE) as part of the Decisional DNA which allows the acquisition and storage of formal decision events in a knowledge-explicit form. It comprises variables, functions, constraints and rules associated in a DNA shape allowing the construction of the Decisional DNA of an organization. Having a powerful knowledge

* Corresponding author. Tel.: +61 2 49217465; fax: +61 2 49217468.

E-mail addresses: Cesar.Sanin@newcastle.edu.au (C. Sanin), ctoro@vicomtech.org (C. Toro), Haoxi.Zhang@newcastle.edu.au (Z. Haoxi), Leonardo.MancillaAmaya@newcastle.edu.au (L. Mancilla-Amaya).

structure such as the Set of Experience Knowledge Structure (SOEKS) within the Decisional DNA can enrich and develop any decisional system based upon previous experience.

This paper presents the Decisional DNA as a multi-domain knowledge structure that provides additional support by constructing a decisional repository, i.e., decisional fingerprints. Additionally, such decisional repository, since it is multi-technology applicable, can be shared and distributed enhancing the user's decisional experience. We present its application into four technologies: Decisional DNA Ontology-based knowledge Structure, Reflexive Ontologies, Embedded Systems, and Decision Support Medical Systems. We have chosen these technologies due to their noticeable advantages of being wide spread technologies that are developing the Artificial Intelligence (AI) scientific field. This paper is organized as follows: In Section 2, an overview of the conceptual basis is presented. In Section 3, we introduce four technologies implementing the Decisional DNA knowledge structure. And in Sections 4 and 5, we present our conclusions and lines for future work.

2. Conceptual basis and background

Humanity has always being accompanied by knowledge. Both have grown together to construct what we understand now as society and civilization. Hence, humankind has been trying to make knowledge part of its assets. Knowledge seems to be a valuable possession of incalculable worth and it has been considered as the only true source of a nation's economic and military strength as well as, the key source of competitive advantage of a company [2]. Thus, humankind in general and, more specifically, managers have turned to knowledge administration. They want technologies that facilitate control of all forms of knowledge because such technologies can be considered as the key for the success or failure of an organization, and subsequently, knowledge society. Knowledge itself appears as a human being attribute and can be defined as [3]: (i) theoretical or practical expertise and skills gained by a person through experience or education, or (ii) familiarity gained by experience of a fact or situation.

One theory suggests that situation assessments are the base for experienced decision-makers when taking decisions [4]. Decision-makers principally use experience for their decisions, i.e., when a decision event emerges, managers select actions that have worked well in previous similar situations. Then, in a brain process that is not well understood yet, managers extract the most significant characteristics from the current circumstances, and relate them to similar situations and actions that have worked well in the past. Therefore, this theory suggests that any mechanism that supports the process of storing previous decisions would improve the decision maker's job; and as such, it is related to a process of storing knowledge and experience.

Since this paper tackles problems in the engineering and computer fields, we concentrate on the concept knowledge engineering (KE). According to Feigenbaum [1], "Knowledge Engineering (KE) is an engineering discipline that involves integrating Knowledge into computer systems in order to solve complex problems normally requiring a high level of human expertise". Two main movements surround KE, they are: the transfer movement and the modeling movement. The transfer movement aims for techniques to transfer human knowledge into the artificial intelligent systems. The modeling movement aims for modeling the knowledge and problem solving techniques of the domain expert into the artificial intelligent system. Our research concentrates on the modeling trend which requires the areas of knowledge representation (KR) and knowledge modeling. Therefore, KE [5] depends on computer science in general, trying

to mimic knowledge in a certain domain and within the scope of an artificial system. This definition involves not only the need for specific technologies, but also the need to overcome related implementation issues.

From a mechanistic point of view, reasoning in machines is a computational process. This computational process, to be feasible, definitely needs systemic techniques and data structures, and in consequence, several techniques have been developed trying to represent and acquire knowledge. These kinds of technologies try to collect and administer knowledge in some manner. Although these technologies work with decision-making in some way, they lack of keeping structured knowledge of the formal decision events they participate on [6]; they do not use their experience. We formally define a *Formal Decision Event* as a choice [decision] made or a commitment to act that was the result [consequence] of a series of repeatable actions performed in a structured manner.

For us, any technology able to capture and store formal decision events as explicit knowledge will improve the decision-making process. Such technology will help by reducing decision time, as well as avoiding repetition and duplication in the process.

Unfortunately, computers are not as clever as to form internal representations of the world, and even simpler, representations of just formal decision events. Instead of gathering knowledge for themselves, computers must rely on people to place knowledge directly into their memories. This problem suggests deciding on ways to represent information and knowledge inside computers.

A Knowledge Representation (KR) is fundamentally a replacement, a substitute for the thing itself. KR is an element of intelligent reasoning, a medium for organizing information to facilitate making inferences and recommendations, and a set of ontological commitments, i.e., an answer of how to interpret the world [7]. KR has been involved in several science fields; however, its main roots come from three specific areas: logic, rules, and frames. They appear as the most generalized techniques, and symbolize the kinds of things that are important in the world; even though developed systems can use exclusively one of the techniques, their hybridization is a common element. Logic implicates understanding the world in terms of individual entities and associations between them. Rule-based systems view the world in terms of attribute-object-value and the rules that connect them. Frames, on the other hand, comprise thinking about the world in terms of prototypical concepts. Hence, each of these representation techniques supplies its own view of what is important to focus on, and suggests that anything out of this focus may be ignored [7]. Recent advances in the field of KR have converged on constructing a Semantic Web, an extension of the current World Wide Web, looking for publishing information in a form that is easily inferable to both humans and machines. Current progresses have led to the standardization of the Web Ontology Language (OWL) by the World Wide Web Consortium (W3C). OWL provides the means for specifying and defining ontologies, that is, collections of descriptions of concepts in a domain (classes), properties of classes, and limitations on properties. OWL can be seen as an extension from the frame based approach to knowledge representation, and a division of logic called Description Logics (DL) [8].

These KR techniques have been implemented with different data structures creating a universe of knowledge as big as the number of applications researchers and IT companies have developed. These technologies have been developed to make useful huge quantities of stored information by modeling knowledge in some way; however, none of them keep an explicit record of the decision events they participate on. Hence, it is necessary to define a multi-domain shareable knowledge structure able to be adaptable and versatile as to capture all these different decision events from the day-to-day operation, to store proper

characteristics of the experience acquired, to keep this experience as explicit knowledge, and to allow it for multiple technologies to be used, analyzed, and categorized.

This paper presents four technologies that use Decisional DNA (and the Set of Experience Knowledge Structure – SOEKS – within it) as the knowledge representation. Decisional DNA is offered as a solution to be utilized for the aims mentioned above. Decisional DNA and SOEKS certainly improve KE and the quality of decision-making by advancing the notion of administering knowledge in the current decision making environment.

2.1. Set of experience knowledge structure (SOEKS) and Decisional DNA

In living species, Deoxyribonucleic Acid (DNA) is a nucleic acid found in cells that carries genetic information, and is the molecular basis of heredity. DNA is made from a combination of four basic elements called nucleotides. These nucleotides are Adenine (A), Thymine (T), Guanine (G) and Cytosine (C). Their combination allows for the different characteristics of each individual, and becomes as one of the highlighted uniqueness of this kind of structure. One part of the long strand comprises a gene. A gene is a portion of a DNA molecule, which guides the operation of one particular component of an organism. Genes give orders to a living organism about how to respond to different stimuli. Finally, a set of genes makes a chromosome, and multiple chromosomes make the whole genetic code of an individual [9]. DNA demonstrates unique aspects as a data structure [10]. Information about the living organism is kept to be passed on to future generations, as well as being the basis of new elements in the organism which are evaluated in terms of performance. DNA stores information for the survival of the species, and improvement in the evolutionary chain.

In our research and taking experience as one of the most valuable ways to acquire knowledge, we rely on computers as in important mean to capture it. However, computers must depend on human beings to enter knowledge directly into them. Thus, the problem is how to adequately, efficiently, and effectively represent information and knowledge inside a computer.

Based upon the DNA concept and using it as a metaphor, we developed the Set of Experience Knowledge Structure (SOEKS) as a form to keep FDE in an explicit way [6]. It is a model based upon existing and available knowledge, which must adjust to the decision event it is built from (i.e., it is a dynamic structure that relies on the information offered by a FDE). Four basic components surround decision-making events and FDE, and are stored in a combined dynamic structure that comprises the SOE (Fig. 1); they are: *variables V*, *functions F*, *constraints C*, and *rules R*.

Variables usually involve representing knowledge using an attribute-value language (i.e., by a vector of variables and values) [11]. This is a traditional approach from the origin of knowledge

representation, and is the starting point for the SOEKS. Variables that intervene in the process of decision-making are the first component of the SOE. These variables are the center root of the structure, because they are the source of the other components.

Based on the idea of Malhotra [12] who states that “to grasp the meaning of a thing, an event, or a situation is to see it in its relations to other things” (p. 51), variables are related among them in the shape of functions. Functions describe associations between a dependent variable and a set of input variables; moreover, functions can be applied for reasoning optimal states, because they come out from the goals of the decision event. Therefore, the SOE uses functions, its second component, and establishes links among the variables constructing multi-objective goals, that is, multiple functions that restrict experience on decision-making by the elements of a universe of relationships.

According to Theory of Constraints (TOC), Goldratt [13] maintains that any system has at least one constraint; otherwise, its performance would be infinite. Thus, constraints are another way of relationships among the variables; in fact, they are functions as well, but they have a different purpose. A constraint, as the third component of SOEKS, is a limitation of possibilities, a restriction of the feasible solutions in a decision problem, and a factor that limits the performance of a system with respect to its goals.

Finally, rules are suitable for representing inferences, or for associating actions with conditions under which the actions should be done. Rules, the fourth component of SOEKS, are another form of expressing relationships among variables. They are conditional relationships that operate in the universe of variables. Rules are relationships between a condition and a consequence connected by the statements IF-THEN-ELSE.

As stated above, SOE is a Knowledge Structure that is able to store and act as a repository of decisional experiences based upon FDE; therefore, functions and operations acting upon such knowledge structure are commonly operated actions performed on traditional computer-based structures. Many of these operations are based upon the specific technology that is applying the SOEKS, for instance, in [14], Sanin and Szczerbicki used SOE in combination with Genetic Algorithms (GA) techniques and therefore, the SOEKS absorbed GA operations such as finding individual fitness, mutations and crossover. Another example takes SOEKS to be implemented on Reflexive Ontologies (RO) [15]; in such case, Toro et al. create a SOEKS RO-based which uses RO operations, allowing SOEs to be operated by Union, Intersection and other set operations. Summarizing, SOEKS as a knowledge structure absorbs operations depending on the technology that applies it.

Furthermore, the SOEKS takes other important features of DNA. Firstly, the combination of the four nucleotides of DNA gives uniqueness to itself, just as the combination of the four components of the SOE offer distinctiveness. Moreover, the four elements of a SOE are connected among themselves imitating part

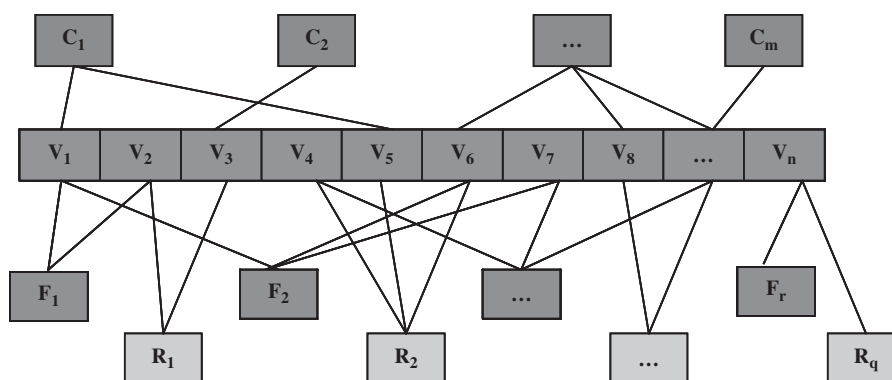


Fig. 1. Set of Experience Knowledge Structure (SOEKS).

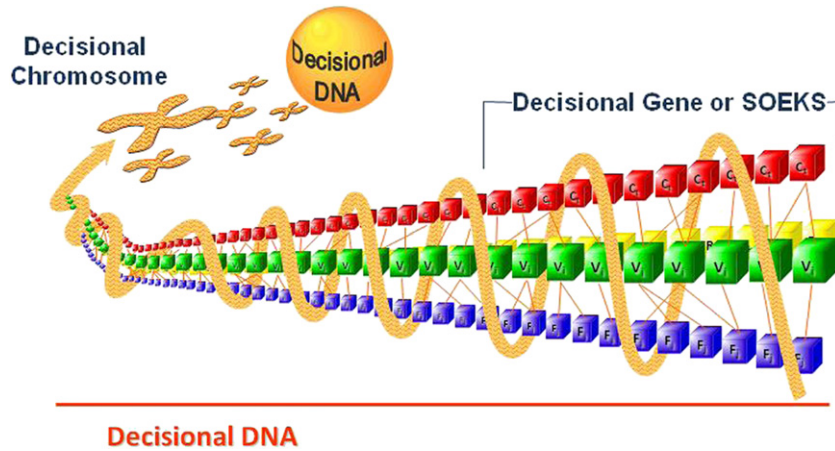


Fig. 2. SOEKS and Decisional DNA.

of a long strand of DNA, that is, a gene. Thus, a gene can be assimilated to a SOE, and in the same way as a gene produces a phenotype, a SOE produces a value of decision in terms of the elements it contains. Such value of decision can be called the efficiency or the phenotype value of the SOE [6]; in other words, the SOEKS, itself, stores an answer to a query presented. Each SOE can be categorized, and acts as a gene in DNA. A gene guides hereditary responses in living organisms. As an analogy, a SOE guides the responses of certain areas of decision making; in our case, a decisional gene.

A unique SOE cannot rule a whole system, even in a specific area or category. Therefore, more Sets of Experience should be acquired and constructed. The day-to-day operation provides many decisions (or FDE), and the result of this is a collection of many different SOE. A group of SOE of the same category comprises a decisional chromosome, as DNA does with genes. This decisional chromosome stores decisional “strategies” for a category. In this case, each module of chromosomes forms an entire inference tool, and provides a schematic view for knowledge inside an organization. Subsequently, having a diverse group of SOE chromosomes is like having the Decisional DNA of an organization, because what has been collected is a series of inference strategies related to such enterprise (Fig. 2).

In conclusion, the SOEKS is a compound of variables, functions, constraints and rules, which are uniquely combined to represent a FDE. Multiple SOE can be collected, classified, and organized according to their efficiency, grouping them into decisional chromosomes. Chromosomes are groups of SOE that can accumulate decisional strategies for a specific area of decision making. Finally, sets of chromosomes comprise what is called the Decisional DNA of the organization [6]. Furthermore, the Decisional DNA can be used in platforms to support decision-making, and new decisions can be made based on it. In this text a concise idea of the SOEKS and the Decisional DNA was offered, for further information [6] should be reviewed.

3. Constructing Decisional DNA

Applications involved in decision making produce myriads of FDE, i.e., decisional experience, their results are, in most cases, analyzed and stored; however, such decisional experience is commonly disregarded, not shared, and put behind [16–19]. Little of this collected experience survives, and in some cases, over time, it becomes inaccessible due to poor knowledge engineering practices or due to technology changes in software, hardware or storage media. Knowledge and experience are lost indicating that

there is a clear deficiency on experience collection and reuse. We suggest that some of the reasons are:

- the nonexistence of a common knowledge–experience structure able to collect multi-domain and multi-technology formal decision events, and
- the nonexistence of a technology able to capture, store, improve, retrieve and reuse such collected experience.

Through our project, we proposed three important elements:

- a knowledge structure able to store and maintain experiential knowledge, that is, the Decisional DNA and the SOEKS,
- a solution for collecting experience that can be applied to multiple technologies from different domains, that is, a multi-technology knowledge structure, and
- a way to automate decision making by using such experience, that is, retrieve collected experience by answering a query presented.

In this paper, we introduce the reader to the three above mentioned elements throughout four different technologies: Decisional DNA Ontology-based Knowledge Structure, Reflexive Ontologies, Embedded Systems – Interactive TV (iTV), and Decision Support Medical Systems for Alzheimer diagnosis. Nevertheless, we would like to add that Decisional DNA is not limited to these technologies and advances are being made with it in several areas. To our knowledge, multiagents systems, web engineering, and robotics are some additional technologies that are currently using Decisional DNA.

3.1. Decisional DNA ontology-based knowledge structure and Reflexive Ontologies (RO)

This section introduces our approach to model and apply Decisional DNA knowledge structure from an ontology perspective. In order to obtain such ontology, Decisional DNA XML-based was taken as the starting point. For a better understanding of Decisional DNA XML-based, the reader should refer to [20]. Afterward, an ontology model process was performed using the Protégé editor [21].

Reflexivity addresses the property of an abstract structure of a knowledge base (in this case, an ontology and its instances) to “know about itself”. When an abstract knowledge structure is able to maintain, in a persistent manner, every query performed on it, and store those queries as individuals of a class that extends the original ontology, it is said that such ontology is reflexive.

Thus, Toro et al. [15] proposed the following definition for a Reflexive Ontology: “A Reflexive Ontology is a description of the concepts and the relations of such concepts in a specific domain, enhanced by an explicit self contained set of queries over the instances”. Therefore, any RO is an abstract knowledge structure with a set of structured contents and relationships, and all the mathematical concepts of a set can be applied to it as a way of formalization and handling. A RO is, basically, an ontology that has been extended with the concept of reflexivity and must fulfill the properties of: query retrieval (storing every query performed), integrity update (updating structural changes in the query retrieval system), autopoietic behavior (capacity of self creation), support for logical operators (mechanisms of set handling), and self reasoning over the query set (capacity of performing logical operations over the query system). The advantage of implementing RO relies on the following main aspects: Speed on the query process, incremental nature, and self containment of the knowledge structure in a single file.

The purpose of this case study is to exemplify how the SOEKS-OWL is converted into a Reflexive Ontology by the use of the ReflexiveQueryStorer class and the changes it generates in such ontology; therefore, we start from a point where a SOEKS-OWL has been already instantiated with real values. Next step comprises the adaptation of the Reflexive Query Storer class with some initial values such as the path of the ontology, options of saving the reflexive structure and the query instances, and the type of query to be performed. For explanation purposes, three queries are included in this case study. The first query is defined in the code as: `public static String SIMPLE_RFLEXIVE_QUERY='CLASS variable with the PROPERTY var_name EQUALS to X1'`; notice that this is a value type query. Such query is written in a human readable form, but in other terms, it means “retrieve all the variables of the ontology that have the variable name X1”.

The execution of the code offers information about the type of query executed and the successful saving of the Reflexive Ontology Structure (created for first time) as well as the query executed with results. Following, the results can be seen as a query successfully executed with the new instance in the SOEKS-OWL transformed into a Reflexive Ontology (Fig. 3):

```
START
Testing Simple query : CLASS variable with the
PROPERTY var_name EQUALS to X1
... saving successful.
File modification saved with 0 errors.
END
```

The next example includes a data type query: `public static String SIMPLE_RFLEXIVE_QUERY='CLASS term with the PROPERTY withVariable EQUALS to X2_1; or in other words it MEANS “retrieve all the terms in the ontology that involve the variable with name X2_1”`. Its results are as follows (Fig. 4):

```
START
Testing Simple query : CLASS term with the PROP-
ERTY withVariable EQUALS to X2_1
... saving successful.
File modification saved with 0 errors.
END
```

In these results can be seen that the term_2 and term_4 are valid for the query; in other words, those terms contain the X2_1 variable. One of the features of the Reflexive Ontologies is that when a new query comes, it is not necessary to perform it again, if it was already done. Finally, RO also allows performing complex queries which comprise autopoietic behavior and support for logical operations, and example is shown below.

```
START
Testing Complex query : CLASS simfactor with the
PROPERTY hasSterm EQUALS to term_1 AND CLASS sim-
factor with the PROPERTY hasSterm EQUALS to term_2
CLASS simfactor with the PROPERTY hasSterm EQUALS
to term_1 AND
CLASS simfactor with the PROPERTY hasSterm EQUALS
to term_2
Sub queries in query 2
... saving successful.
File modification saved with 0 errors.
END
```

As it has been shown, the Reflexive Ontology transformation includes the creation of a new class inside the ontology, in this case the SOEKS-OWL. Additionally, when different simple or complex queries (data type or value type) are executed, they are inserted as instances in the new Reflexive Ontology; this change facilitates the application of similarity elements among the Sets of Experience and it will allow an extended logic handling over the SOEKS as it comprises the self reasoning over the query set property of the RO.

A RO Decisional DNA-based model, once instantiated, can be accessed through different queries, which would be developed according to similarity parameters [22] and users' requirements.

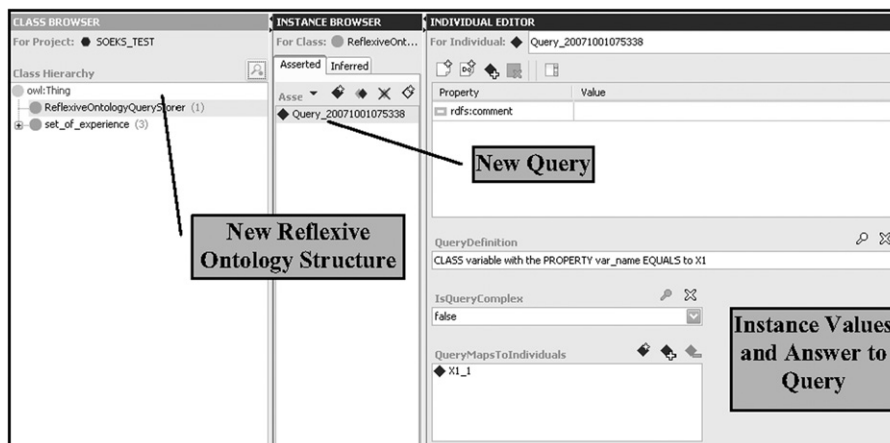


Fig. 3. SOEKS-OWL transformed into a Reflexive Ontology with its new elements.

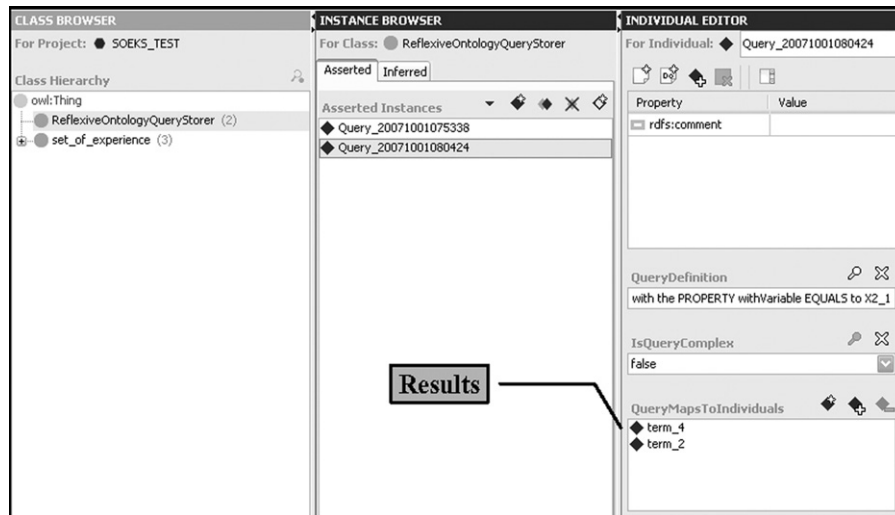


Fig. 4. SOEKS-OWL with a data type query executed.

A powerful representation, querying and inference capabilities are exploited in several ways. For instance, the ability to perform advanced queries on large sets of information with an optimal response time was exploited by means of Reflexive Ontologies (RO) [15,23].

As expressed by Nguyen [24], knowledge of a group is more appropriate than the individual knowledge; therefore, having an ontology-based repository ready to be feed with Decisional DNA produced by members of a decisional community is the beginning of a new way of sharing knowledge. The decisional community would share decisions among its members allowing decision-maker users to improve their day-to-day operation by querying such repository, and along with this interaction, the decisional community would increase and improve the Decisional DNA available for being shared.

3.2. Decisional DNA-based embedded systems

Embedded systems are computer systems created to implement one or a few dedicated functions [25] such as the control system in an elevator or the ABS (anti-lock braking systems) in a car. They are usually embedded as part of a complete system. Now-a-days, any device that includes a computer, but it is not a general-purpose computer itself, can be regarded as an embedded system [26]. ES range from household appliances like microwave ovens and washing machines, to industrial applications like automatic production lines and network switches; from portable devices such as MP3 players to very big equipments like nuclear power plants. Recent advances in microelectronics, IC (integrated circuit), communications, computing, software and other information technologies are influencing ES to be increasingly powerful and popular than ever. Such technologies are enabling ES to be built as what is required to meet all kinds of demands from our daily life [27].

In addition to embedded operating systems, software/firmware and middleware technologies for ES are making notable progress as well. Thanks to improvements on software, hardware and embedded operating systems, it is now possible to apply more mature approaches onto ES, like using regular Java, C or C++ compiler on the ES itself, making applications portability an easier job [27].

Knowledge-based embedded systems are the new trend on Embedded Systems (ES) and they can make our world increasingly intelligent and convenient. However, due to lack of

standardized solutions and development platforms, it is very hard to share knowledge among different knowledge-based systems. Therefore, we present Decisional DNA-based embedded systems as an alternative for knowledge-based developments in this area which can potentially be applied on various systems and enable them with acquisition, reusing, evolving and sharing knowledge.

In order to use Decisional DNA on embedded systems, a series of required features were defined; they involved a transformation on the traditional Decisional DNA in order to make it a viable alternative for ES. Decisional DNA was then adapted to make it *adaptable and cross-platform portable* (not a problem since the Decisional DNA API is already written in Java), *Compact* and *Configurable*.

3.2.1. Interactive TV Decisional DNA-based software architecture

Interactive television (also known as iTV) is an evolutionary integration of the Internet and Digital TV (DTV) [28]. It contains a number of novel smart technologies and enables viewers to interact with television services and content. The most exciting thing about an interactive TV is the ability to run applications that have been downloaded as part of the broadcast stream. For the user (viewer), this is what really makes a significant difference between a basic Digital TV box and an interactive TV system. In order to support and enable interactive applications, the receiver is required to support not only the implementation of APIs (Application Programming Interface) needed to run the applications, but also the infrastructure needed to inform the receiver what applications are available and how to run them.

Decisional DNA, as a domain-independent, flexible and standard knowledge repository, can not only capture and store experiential knowledge in an explicit and formal way, but can also be easily applied to various domains to support decision-making and standard knowledge sharing and communication among these systems [6] [29]. In this paper, we present an approach that integrates Decisional DNA with iTV to capture and reuse viewers' TV watching experiences. We have demonstrated this approach in order to test the usability and suitability of Decisional DNA in ES.

Decisional DNA iTV consists of the User Interface, the System I/O (input/output), the Integrator, the Prognoser, the Converter and the Decisional DNA Repository (Fig. 5).

User Interface: The User Interface is developed to interact with the user/viewer. In particular, the user can control, set and

configure the system, select services, give feedback, and interact with the service providers by using the User Interface.

System I/O: The System I/O allows the Decisional DNA iTV platform communicating with its domain of operation. The System I/O tells the iTV which service is selected, for example, what movie should play or what feedback was given, etc. Additionally, it is able to access and retrieve the media stream, viewers' feedback, system time, and service information from its domain.

Integrator: The Integrator is where the scenario data is gathered and organized. In our case, we link each decisional experience with a certain scenario describing the circumstances under which the experience is collected, such as the system time, name of a selected service, user input, and other service information. The Integrator organizes the scenario data and sends them to the Prognoser for further processing.

Prognoser: The Prognoser is in charge of sorting, analyzing, organizing, creating and retrieving collected experience. It sorts

data received from the Integrator to further analyze and organize it according to system configurations. Finally, it interacts with the Decisional DNA Repository and the XML Parser in order to store and reuse experience depending on the purpose of different tasks.

Converter: The Converter translates knowledge statements generated by the Prognoser into the Decisional DNA structure and interprets the retrieved Decisional DNA experiences for future reusing. In this case, we used Decisional DNA XML-based.

Decisional DNA Repository: The Decisional DNA Repository is the core architecture component in the iTV approach. It is the place where experiences are stored and managed. It contains the Repository Manager which is the interface of the Decisional DNA Repository. It answers operation commands sent by the Prognoser and manages the Chromosomes.

3.2.2. iTV case study

At this stage, the main purpose of our experiments was to prove that Decisional DNA could be implemented with Java TV providing its domain with the ability of experience capturing and reusing. For testing the concept of Decisional DNA applied to iTV, we used Java TV SDK on a generic setup box (Digitel+HD3000) and considered only five types of movies, namely action, adventure, animation, comedy, and crime. We simulated viewers watching movies on the Decisional DNA iTV, where each movie was represented by its type and an ID number, for example Action1, Comedy2, and with 20 movies/type making a pool of 100 movies.

Fig. 6 shows a screenshot of the viewers' TV. As it can be seen, the viewers' screen is composed of five components: Service Name which shows "Movies", Service Information which displays information about a selected movie, Ranking of the selected movie, Movie Showcase which shows movies recommended by the system, and "Show More ..." button where the viewer can access additional movies. Initially, Decisional DNA iTV recommends two movies from each movie type creating an initial selected group of 10 movies. Once the system start collecting experiences, it recommends movies according to those experiences.

We capture viewers' watching experiences by collecting movie and user's knowledge and information, that includes rules relating preferences, times, dates, actors, ranking, director, among others. Such knowledge is gathered and organized by the Integrator to be sent to the Prognoser, which finally, transforms and stores them into a SOEKS XML format. Once the system captures such experiences, it begins to analyze the viewers' watching

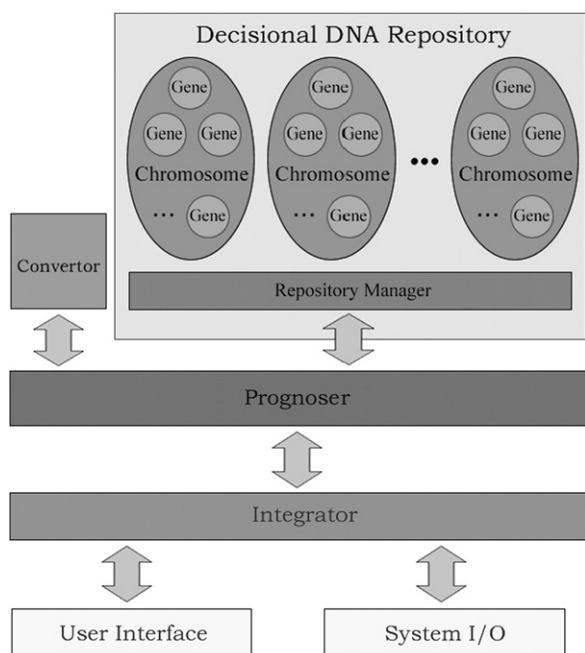


Fig. 5. System Architecture for Decisional DNA iTV.

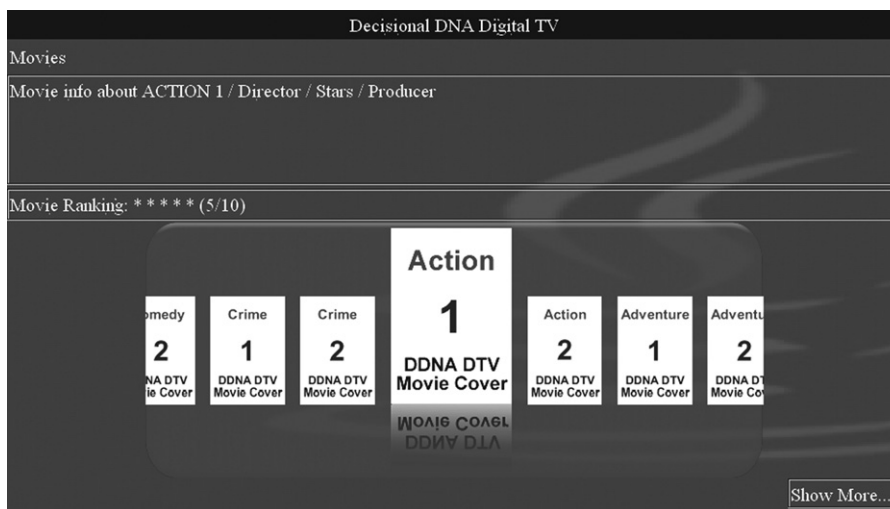


Fig. 6. Screenshot of Decisional DNA iTV.

experiences and provides viewers with smart recommendations based on his/her past viewing experiences.

During the process in which the Prognoser recommends new movies to the viewer, it retrieves watching experiences stored in the Decisional DNA Repository and analyses those experiences from the perspective of the user's settings. In our experiments, we analyze the movie types, most frequent week day and time, and preferences collected in the Decisional DNA iTV for each viewer. At this stage, simple mathematical model have been implemented; however, more advanced mathematical models are in our future research agenda. For instance, Eq. (1) demonstrates how the system calculates the number of movies that should be considered for the movie recommendation list:

$$N = (T \times 100/D + 5)/10 \quad (1)$$

where N represents number of movies that should be recommended from a specific movie type; T tells us how many movies of a specific movie type have been watched on a given week day; D represents the total number of movies watched on that given week day. For example, suppose that there is a viewer who watched 11 movies in total during a given weekend and 5 of those movies were action movies. Therefore, according to (1) there should be five out of ten action movies in the next recommendation list: $(5 \times 100 / 11 + 5) / 10 = 5$. During a number of weeks of capturing experience, the system learns and remembers that this viewer watched on average five action movies, two adventure movies, one animation movie, two comedy movies, and one crime movie on weekends so far. As the result, the system would recommend for this viewer similar a movie composition on the next Friday for the following weekend. Fig. 7 shows the screenshot of a newly recommended movie list in this case.

We presented an approach that integrates Decisional DNA with iTV allowing capturing and reusing viewers' TV watching experiences. Further research involves query enhancement, refinement and further development of the Prognoser algorithms, and implementation of more advanced ways to interpret the viewer experience.

3.3. Clinical decision support system Decisional DNA-based

The interest of making Clinical Decision Support Systems (CDSS) for the diagnosis of Alzheimer Disease (AD) is huge, as it is the leading cause of dementia in developed countries. Early diagnosis of AD is commonly carried out analyzing results of different medical tests, which are multidisciplinary by nature,

such as neurological, neuropsychological, and neuroimaging tests [30]. During this process, large amounts of parameters are generated and making a proper diagnosis becomes a knowledge handling problem. CDSS help physicians overcome knowledge handling problems that they face in their work. During diagnosis processes, CDSS analyze data from medical tests and present results to physicians such that they can diagnose properly in a easily and efficiently from.

In this section, we present a CDSS that (i) supports physicians during diagnosis of AD and (ii) offers tools needed to fulfill the aforementioned need of discovering relevant parameters for this diagnosis. In fact, this CDSS is based on the experience acquired or learned from the user, and it enables the discovery of new knowledge in the system and the generation of new rules based on experience that drive the reasoning. Among several approaches that can be used to endow the proposed system with the ability of adapting and discovering rules when special conditions are encountered, we have chosen the Set Of Experience Knowledge Structure (SOEKS) and the Decisional DNA (DDNA) [6] [29] in their OWL form [31] as a novel way of attaining this behavior. These elements will allow the system to capture previous experiences and discover new knowledge using bio-inspired techniques and the reasoning capabilities offered by ontologies, for instance, the fast query systems presented by Toro et al. [15].

In our system, the experience of the physician while using our system is stored and, with this experience, the system is able to (i) *make explicit the implicit knowledge contained in the system* and (ii) *generate new criteria to drive reasoning*. Supporting our system are widely used ontologies within the medical domain; they are: the Semantic Web Application in Neuromedicine (SWAN) [32] and the Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT) [33].

The CDSS system is based on Decisional DNA OWL based for the knowledge representation and a semantic reasoning process that inferred diagnoses for patients. The semantic reasoning was driven by a static set of production rules provided by AD experts. Its architecture consists of 5 layers: Data Layer, Translation Layer, Ontology and Reasoning Layer, Experience Layer and Application Layer. Among these layers, the Experience Layer is based on the SOEKS and the Decisional DNA, and it is in charge of storing users' experiences (the methodology and criteria used for the diagnosis process), in forms that represent FDE in an explicit way. This experience is then applied, and new knowledge and new rules that drive the diagnosis are discovered by the system. In this way,

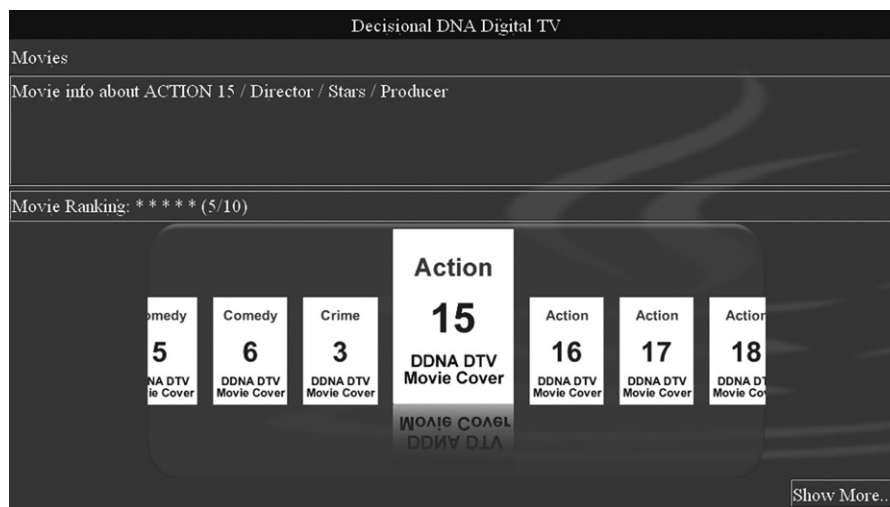


Fig. 7. Newly recommended movie list for the experimental viewer.

physicians are suggested not only diagnoses but also new or modified rules to achieve those diagnoses.

The implementation of the CDSS Decisional DNA-based is being developed as part of the Spanish MIND project (<http://www.portalmind.es>), which follows a multidisciplinary approach for the early detection of AD. Clinical trials are being performed on more than 350 patients in three hospitals of Valencia (Spain), with the intention of gathering information about the early diagnosis of AD.

3.3.1. Evolving the set of rules by the means of Decisional DNA

As a type of decision-makers, medical experts also base their current decisions on lessons learnt from previous similar situations, which in the context of Alzheimer diagnosis are represented by studies performed on several groups of patients under different contexts. In spite of the wide range of scenarios considered by medical studies, the rules and conditions that derived from them may prove to be insufficient, too general, or simply not relevant in scenarios with very particular characteristics. This situation clearly illustrates the need for an automated solution capable of determining adaptability in the set of rules of the diagnosis system, with the purpose of increasing the accuracy and effectiveness of the diagnoses made by medical experts. The use of Decisional DNA takes existing decisions made by experts stored in the system and feed them into a SOEKS/DDNA ontology. Additionally, each decision is translated into its corresponding SOEKS equivalent, and then the system will be able to infer new rules in three categories:

- Fine tuned rules: combination of existing rules to generate a new one.
- Deprecated rules: rules are deemed not to be relevant anymore based on previous experiences.
- Original rules: rules discovered by the system which were not apparent to the experts.

In order to successfully accomplish the system's aim, some considerations were taken into account. Firstly, rules in the Ontology and Reasoning layer are defined by experts, in other words, they are heuristics representing the experience of several medical practitioners, which means they are decisions and when translated into SOEKS, they were considered as FDE. Second, for all knowledge stored in the ontologies, several restrictions on variables possible values have been defined, however, not mathematical functions that relate the different variables in an independent/dependent form were defined.

Initially, the system requires data from the different trials performed on the patients. Such data is gathered via a web-based system called ODEI. When new data is loaded, the MIND Ontology is instantiated using the information provided by users through ODEI's user interface. Then, a semantic reasoning process based on the initial set of production rules is executed with the objective of inferring diagnoses. An evaluation of the inferred diagnoses and decisions on the appropriate course of action are made by the physicians; their final decisions are loaded to the Decisional DNA ontology. As described previously, a translation and inference process between the SOEKS/Decisional DNA ontology is required. However, performing such translation process on a 1-on-1 basis is not practical, it is time consuming with a large number of concurrent users, and may lead to inaccurate results when the system is "learning" (i.e., has little or no experiences in its initial state). This last issue is caused because an accurate inference requires the evaluation of similar elements or situations; therefore, numerous experiences are preferred in order to execute the automated inference process. Consequently, a

micro-batch approach is proposed, similarly to the ones used in data warehouses that allows processing a reasonable amount of data without the heavy workload of large batch processes, or the inherent infrastructure complexity required for real-time or near real-time processing. Additionally, processing small batches of knowledge allows the system to deliver better inference results even when the system is still "learning". The batch process loading the Decisional DNA ontology has two main steps, translating and inferencing. The translation process uses a parser in charge of reading the knowledge from the MIND ontology, extracting the details of all OWL classes, individuals and attributes and inserting them into the SOEKS-OWL form using the SOEKS API. This API is a Java-based library that provides the means to create manipulate and import/export SOEKS in XML or OWL formats. The parser comprises different sub modules, each module creates an image in memory of the SOEKS that is being processed, which is written to the Decisional DNA ontology once the extraction is finished. To illustrate the functionality of the modules, we use an example production rule. It is assumed that the variables and restrictions in the following example are already stored in the MIND ontology:

```
IF((CLASS NeuropsychologicalInformation WITH THE
PROPERTY NeuropsychologicalInformation_FAQPfeffer
GREATER THAN 5)) AND (CLASS NeuropsychologicalInformation
WITH THE PROPERTY NeuropsychologicalInformation_GDS
SMALLER THAN 6)
THEN (( CLASS Diagnosis WITH THE PROPERTY Diagnosis_ReasonedDiagnosis
EQUALS TO ProbableAlzheimer ) AND (CLASS Diagnosis WITH THE PROPERTY Diagnosis_ReasonedRisk
EQUALS TO Low) )
```

In the first place, the class module reads every class in the MIND ontology and translates them into individual SOEKS. In the example, we have the classes NeurophysycologicalInformation and Diagnosis; as a result, two SOEKS instances (i.e., two experiences) are created, as follows:

```
SOEKS NeurophysycologicalInformation =new SOEKS();
Category cat=new Category();
cat.setArea('Neuro Psychological Information');
NeurophysycologicalInformation.setCategory(-cat);
SOEKS Diagnosis =new SOEKS();
cat.setArea('Diagnosis');
Diagnosis.setCategory(cat);
```

Each of these experiences has different variables. For the NeurophysycologicalInformation class, the variables are FAQPfeffer and GDS, and for the Diagnosis class, the variables are ReasonedDiagnosis and ReasonedRisk; therefore, the variable module creates two variables as shown below:

```
Variable FAQPfeffer =new Variable('FAQPfeffer',Variable.VARIABLE_TYPE_NUMERICAL,causeValue,
effectValue,unitType,true);
Variable GDS =new Variable('GDS',Variable.VARIABLE_TYPE_NUMERICAL,causeValue,
effectValue,unitType,true);
Variable ReasonedDiagnosis =new Variable('ReasonedDiagnosis',Variable.VARIABLE_TYPE_CATEGORICAL,
causeValue,effectValue,unitType,true);
Variable ReasonedRisk =new Variable('ReasonedRisk',Variable.VARIABLE_TYPE_CATEGORICAL,
causeValue,effectValue,unitType,true);
```

Once the SOEKS and its variables are created, the constraints module will read the OWL properties and constraints for every variable and construct the constraints elements in memory. For example, according to the production rule, `FAQPfeffer` is greater than 5; therefore a constraint based on this knowledge should look like this:

```
Constraint FAQ_Constraint=new Constraint();
FAQ_Constraint.value(5);
FAQ_Constraint.symbol('>');
FAQ_Constraint.variable(FAQPfeffer);
```

This process is repeated for every constraint and variable in the system. The last step before inserting the experience into the Decisional DNA ontology is to link all the elements of each SOEKS together. To do this, we create the different sets of the SOEKS. The following code fragment illustrates the process with the `NeuropsychologicalInformation` SOEKS.

```
VariableSet varSet=new VariableSet();
varSet.add(FAQPfeffer);
varSet.add(GDS);
NeuropsychologicalInformation.setSetOfVariables(varSet);
ConstraintSet consSet=new ConstraintSet();
consSet.add(FAQ_Constraint);
NeuropsychologicalInformation.setSetOfConstraints(consSet);
```

Finally, the translation process writes the SOEKS in an OWL ontology. This is done by simply calling the `soeksToOWL()` method provided by the SOEKS API. After all the experiences in the batch are translated using the ideas described before, the inference process is executed to discover new rules according to the categories described above. Then, assuming the existence of more knowledge in the system, under specific conditions and after validation against other experiences, the inference process is able to determine that the values obtained from the Folstein test and the probabilities of suffering from AD are related. As a result, the original rule discovered by the system is as follows:

```
IF ((CLASS NeuropsychologicalInformation WITH THE
PROPERTY NeuropsychologicalInformation_MMSEfolstein
SMALLER THAN 16))
THEN ( CLASS Diagnosis WITH THE PROPERTY Diagnosis_ReasonedDiagnosis
EQUALS TO ProbableAlzheimer)
```

As a result of the extension of the system with the Experience Layer, using SOEKS and Decisional DNA, the system is now able to discover new knowledge and rules using bio-inspired techniques, and the reasoning capabilities offered by ontologies. By using these methods, the system acts as an advisor for physicians and supports their decisions.

4. Conclusions

Along this paper, we have focused on the technical elements required to implement Decisional DNA and SOEKS on different technologies and showed that these novel knowledge representations can be shared or interpreted in several ways across different domains. Additionally, we present different conceptual strategies that might be used to exchange knowledge represented as Decisional DNA and SOEKS, with the goal of supporting complex decision-making processes by autonomous and intelligent means, regardless

of the underlying technology. Once Decisional DNA is collected, the possibilities are increased if converted into a standard language such as XML or OWL. Thus, Decisional DNA can offer transportability and shareability characteristics, and therefore, collected experience can be reused in different systems that conform to the Decisional DNA specifications. Additionally, once the Decisional DNA is constructed, the experience repository acts as an enhancing tool for different systems by adding predicting capabilities and facilitate knowledge engineering processes inside decision making.

Decisional DNA and SOEKS can indeed improve the current state of the art of Knowledge Engineering Applications. The benefits of using the Decisional DNA are evident; however, we believe that some challenges are still open. Some of such challenges are the testing of Decisional DNA in more technologies as well as on multimedia applications in a way that can collect FDE related to images and sound.

Decisional DNA enables us to distribute experience among different applications, and in that form, and through a decisional community, organizations that are expanding the knowledge management concept externally, can explore new ways to put explicit classifiable knowledge in the hands of employees, customers, suppliers, and partners.

5. Future work

In terms of software agents, our intention is to develop a platform that takes into account the aforesaid ideas and concepts as the driving force that will help us create a knowledge market using SOEKS and *Decisional DNA*. There have been some work and experiments as first steps toward the development of such platform, and further refinement, validation and testing is being carried out. Another topic in the future research agenda for the *E-Decisional Community* is *Decisional DNA* appropriation, which means using the elements defined above we plan to determine how new knowledge is merged with the one that an entity already possesses. Determining the quality of knowledge and user satisfaction, among other criteria, will support this process. Desirably, once agents have assimilated knowledge after several interactions, the answer of a query should eventually converge to a common and globally accepted solution.

In regards to embedded systems, we are using robotics as test platforms. We have worked in an approach that allows a robot to capture and reuse its own experiences by applying Decisional DNA. Since the Decisional DNA applied to robotics is at its early development stages, there are further research and refinement remaining to be done, some of them are: Refinement of the requirements of Decisional DNA for robotics, such as inter-process communication strategies, life cycle management and protocols need to be explored in detail; enhancement of the compactness and efficiency of Decisional DNA: optimization of algorithms, evaluation and comparison of the different strategies for Decisional DNA repository storage and query; and, technical review of distributed systems and middleware to determine their viability for a future implementation of the Decisional DNA applied to robotics. Including shareability of gained experience among different robots.

References

- [1] E. Feigenbaum, P. McCorduck, *The Fifth Generation*, Addison-Wesley, London, 1983.
- [2] P. Drucker, *The Post-Capitalist Executive: Managing in a Time of Great Change*, Penguin, New York, 1995.
- [3] Oxford University Press, *Oxford Dictionary of English*, <<http://www.askoxford.com>>, Accessed Oct 2010.
- [4] D. Noble, Distributed situation assessment, in *Proc. FUSION '98*, P. H. R. Arabnia (Ed.), University of Georgia, Athens, 478-485, 1998.

- [5] W. Arnold, J. Bowie, *Artificial Intelligence: A Personal Commonsense Journey*, Prentice Hall, New Jersey, 1985.
- [6] C. Sanin, E. Szczerbicki, Experience-based knowledge representation SOEKS, *Cybernet Syst.* 40 (2) (2009) 99–122.
- [7] R. Davis, H. Shrobe, P. Szolovitz, What is a knowledge representation? *AI Mag.* 14 (1) (1993) 17–33.
- [8] B. Schatz, G. Mohay, A. Clark, Rich Event Representation for computer forensics, in *Proc. 5th Asia Pacific Industrial Engineering and Management Systems Conference – APIEMS, QUT, Gold Coast (2004)* 12–15.
- [9] K.R. Miller, J. Levin, *Biology*, Prentice Hall, Saddle River, 2002.
- [10] W. Ryu. DNA Computing: A Primer, <www.arstechnica.com/reviews/2q00/dna/dna-1.html>, Accessed Nov 10, 2010.
- [11] J.W. Lloyd, *Logic for Learning: Learning Comprehensible Theories from Structure Data*, Springer, Berlin, 2003.
- [12] Y. Malhotra, From information management to knowledge management: beyond the 'hi-tech hidebound' systems, in: K. Srikantaiah, M.E.D. Koenig (Eds.), *Knowledge Management for the Information Professional*, Information Today Inc, 2000, pp. 37–61.
- [13] E.M. Goldratt, J. Cox, *The Goal*, Grover, Aldershot-Hants, 1986.
- [14] C. Sanin, E. Szczerbicki, Genetic algorithms for Decisional DNA: solving sets of experience knowledge structure, *Cybernet Syst.* 38 (5) (2007) 475–494.
- [15] C. Toro, C. Sanin, E. Szczerbicki, J. Posada, Reflexive Ontologies: enhancing ontologies with self-contained queries, *Cybernet Syst.* 39 (2) (2008) 1–19.
- [16] S. Blakeslee, Lost on earth: Wealth of Data Found in Space, *New York Times*, March 20, (1990) C1.
- [17] L. Corti, G. Backhouse, Acquiring qualitative data for secondary analysis, *Forum: Qualitative Social Res.* 6 (2005) 2.
- [18] C. Humphrey, Preserving research data: a time for action, in *Proc. Preservation of Electronic Records: New Knowledge and Decision-making: postprints of a conference – Symposium 2003*, Canadian Conservation Institute, Ottawa (2004) 83–89.
- [19] P. Johnson, Who you gonna call? *Technicalities* 10 (4) (1990) 6–8.
- [20] C. Sanin, E. Szczerbicki, Using XML for implementing set of experience knowledge structure, in *Proc. Int. Conf. on Knowledge-base and Intelligent Information and Engineering Systems – KES, Part I LNCS 3681*, Springer (2005) pp. 946–952.
- [21] Protégé, Stanford Medical Informatics, <<http://protege.stanford.edu/index.html>>, (2005).
- [22] C. Sanin, E. Szczerbicki, Similarity metrics for set of experience knowledge structure, in *Proc. Int. Conf. on Knowledge-base and Intelligent Information and Engineering Systems – KES, Part I LNAI 4251*, Springer (2006) 663–670.
- [23] C. Sanin, E. Szczerbicki, A Decisional Trust Implementation on a maintenance system by the means of Decisional DNA and Reflexive Ontologies, in *Proc. International Conference on Web Intelligence and Intelligent Agent Technology – WI-IAT '08 IEEE/WIC/ACM*, Springer, Sydney (2008) 5–8.
- [24] N.T. Nguyen, Inconsistency of knowledge and collective intelligence, *Cybernet Syst.* 39 (6) (2008) 542–562.
- [25] S. Heath, *Embedded Systems Design*, EDN Series for Design Engineers, 2nd edn., Newnes, 2003.
- [26] W. Wolf, *Computers as Components*, Morgan Kaufmann, San Francisco, 2001.
- [27] C.E. Pereira, L. Carro, Distributed real-time embedded systems: recent advances, future trends and their impact on manufacturing plant control, *Annu. Rev. Control* 31 (1) (2007) 81–92.
- [28] E. Schwalb, *iTV Handbook: Technologies & Standards*, Prentice Hall, Upper Saddle River, NJ, 2003.
- [29] C. Sanin, C. Decisional DNA and the Smart Knowledge Management System: Knowledge Engineering and Knowledge Management applied to an Intelligent Platform, LAP Lambert Academic Publishing, Berlin, 2010.
- [30] B. Monien, L. Apostolova, G. Bitan, Early diagnostics and therapeutics for Alzheimer's disease – how early can we get there? *Expert Rev. Neurother* 6 (2009) 1293–1306.
- [31] C. Sanin, C. Toro, E. Szczerbicki, An OWL ontology of set of experience knowledge structure, *JUCS (J. Universal Comput. Sci.)* 13 (2) (2007) 209–223.
- [32] P. Ciccarese, E. Wu, G. Wong, M. Ocana, J. Kinoshita, The SWAN biomedical discourse ontology, *J Biomed. Inf.* 41 (2008) 739–751.
- [33] M. Nyström, A. Vikström, G.H. Nilsson, H. Ahlfeldt, H. Orman, Enriching a primary health care version of ICD-10 using SNOMED CT mapping, *J Biomed. Semant.* 1 (7) (2010).



Dr. Cesar Sanin has been working in the field of knowledge engineering and intelligent technologies for the past 8 years. He obtained his Administrative Engineering Degree (2000) from the National University of Colombia and a Diploma in IT (2003) at The University of Newcastle, Australia. Afterward, he pursued a PhD degree at the School of Mechanical Engineering of The University of Newcastle, and received his degree in the field of Knowledge Engineering and Intelligent Technologies (2007). Currently, he continues his work at The University of Newcastle as a co-director of the Knowledge Engineering Research Team – KERT. His research focuses on the areas of

knowledge engineering, decision support systems and intelligent systems for engineering and business.



Dr. Carlos Toro received his PhD in Computer Science from the University of the Basque Country in the field of applied Semantics and his diploma degree in Mechanical Engineering with honours from the EAFIT University (Colombia). In 1998, he was invited as student researcher to the LSFA (Large Scale Flexible Automation Lab) at the University of Illinois at Urbana-Champaign USA under the supervision of P.M. Ferreira working in the field of virtual reality and industrial plant layout design. He was appointed lecturer at EAFIT University in CAD/CAM Systems and Conceptual Design between 2003 and 2004. Since 2003 he is a senior researcher at VICOMTech within the Industrial

Applications division where he works in the application of Semantic Technologies in Virtual Engineering leading different national and international projects. He has several scientific publications in conferences and journals in areas such as ontology support to design review and virtual reality walkthroughs, STEP based automatic categorization of 3D CAD elements, concurrent engineering, etc. In 2007 and 2011, he was invited researcher at The University of Newcastle (NSW-Australia), working in the application of semantic technologies and in specific in the enhancement of the ontology query process. Dr. Toro is a member of KES international where he contributes as reviewer and member of the editorial committee. He is an ASME Member (American Society of Mechanical Engineers).



Haoxi Zhang is a PhD student at The University of Newcastle, Australia. After graduating from Computer Science and Technology from the China West Normal University, he worked at Kangding Nationality Teacher's College as a lecturer in the Faculty of Computer Science. Afterward, he started a Master of Software Engineering at the University of Electronic Science and Technology of China, obtaining his Master's degree in 2008. His interests focus on building embedded intelligent systems based on knowledge engineering.



Eider Sanchez is a telecommunications engineer currently pursuing her master degree. She is working as a researcher at VicomTech Institute, Spain in areas associated to knowledge engineering and semantics.



Professor Edward Szczerbicki has had very extensive experience in the area of intelligent systems development over an uninterrupted 25 year period, 8 years of which he spent in top systems research centres in the USA, UK, Germany and Australia. In this area, he contributed to the understanding of information and knowledge management in systems operating in environments characterised by informational uncertainties. He has published close to 300 refereed papers with more than 350 citations over the last 20 years. His DSc degree (1993) and the Title of Professor (2006) were gained in the area of information science for his international published contributions.

Professor Szczerbicki serves as a Board Member of International Academic Advisory Council for Natural and Artificial Intelligence Systems Organization (NAISO), Canada. He has given over 30 invited presentations and addresses at universities in Europe, USA and at international conferences. He is a Member of the Editorial Board/Associated Editor for 8 international journals. He chaired/co-chaired and acted as a committee member for numerous international conferences. He is a General Chair of International Symposium on Advances of Intelligent Systems and an Australian Chairman of an International Conference on Intelligent Processing and Manufacturing. He is a Vice Chair of the IFIP Working Group 7.6 on *Complex Systems in Policy and Management* and a Member of Berkeley Initiative in Soft Computing Special Interest Group on *Intelligent Manufacturing*.

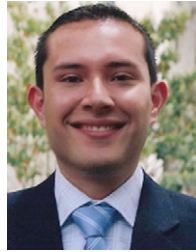


Eduardo Carrasco is a graduate of the San Sebastian Higher School of Industrial Engineers where he read Electronic and Control Systems Engineering. From 1998 until 2000 Eduardo held the position of Assistant Researcher in the Environmental Engineering Unit at CEIT, Centro de Estudios e Investigaciones Técnicas de Gipuzkoa (Guipuzcoa Centre for Studies and Research Techniques). During the same period Eduardo was an assistant lecturer at the San Sebastian Higher School of Industrial Engineers.

In 2001, he held the position of staff Researcher at the Fraunhofer Institut für Graphische Datenverarbeitung (Fraunhofer Institute for Computer Graphics)

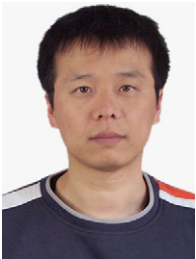
under the Supervision of Dr. Georgios Sakas. During this period his research field was related with the development of an algorithm for realistic 3D object reconstruction from varied 2D images.

From 2003 onward Eduardo has been a Researcher at VICOMTech leading Ambient Intelligence and Accessibility projects.



Leonardo Mancilla-Amaya is a PhD student at The University of Newcastle, Australia. Received his BSc in Computer Science from the Pontificia Universidad Javeriana (Colombia), where he graduated with Honours in 2005. His degree project was focused on the area of Software Agents and Computer Supported Cooperative Work (CSCW). In 2006, he started a Masters by Research in the area of Grid Computing at the Universidad de Los Andes (Colombia), and received his degree in 2008. He has worked in different software companies as a developer, and as a consultant.

He also has been involved in different research projects sponsored by the Colombian Government, based on Software Agents and Robotics. In 2009, he became a PhD student. His interests focus on building intelligent systems associated to agent technologies based on knowledge engineering, large scale distributed systems and information management.



Peng Wang is currently a Ph.D. student at the University of Newcastle, Australia. After graduating as a Computer Scientist in 1998 from Northeastern Normal University, he worked for 9 years in one of the world top 100 companies, China telecommunication Co., in Shanghai. His experience in different technologies and project manager took him to pursue Master studies in Information Technology at The University of Newcastle, graduating in 2010. Currently, his interests focus on discovering abilities of web data mining techniques DDNA-based.