Lima L., Novais P., Costa R., Bulas Cruz J., Neves J., Group Decision Making and Quality-of-Information in e-Health Systems, Logic Journal of the IGPL, Oxford University Press, Volume 19 Issue 2, pp 315-332, ISSN 1367-0751, 2011.

Group Decision Making and Quality-of-Information in e-Health Systems

Luís Lima[1], Paulo Novais[2], Ricardo Costa[1], José Bulas Cruz[3] and José Neves[2]

[1]College of Management and Technology - Polytechnic of Porto, Felgueiras, Portugal
[2]Departamento de Informática/CCTC, Universidade do Minho, Braga, Portugal
[3]University of Trás-os-Montes e Alto Douro, Vila Real, Portugal
lcl@estgf.ipp.pt; pjon@di.uminho.pt; rcosta@estgf.ipp.pt; jcruz@utad.pt; jneves@di.uminho.pt

Abstract. Knowledge is central to the modern economy and society. Indeed, the knowledge society has transformed the concept of knowledge and is more and more aware of the need to overcome the lack of knowledge when has to make options or address its problems and dilemmas. One`s knowledge is less based on exact facts and more on hypotheses, perceptions or indications. Even when we use new computational artefacts and novel methodologies for problem solving, like the use of Group Decision Support Systems (GDSS), the question of incomplete information is in most of the situations marginalized. On the other hand, common sense tells us that when a decision is made it is impossible to have a perception of all the information involved and the nature of its intrinsic quality. Therefore, something has to be made in terms of the information available and the process of its evaluation. It is under this framework that a Multi-valued Extended Logic Programming language will be used for knowledge representation and reasoning, leading to a model that embodies the Quality-of-Information (QoI) and its quantification, along the several stages of the decision making process. In this way it is possible to provide a measure of the value of the QoI that supports the decision itself. This model will be here presented in the context of a GDSS for VirtualECare, a system aimed at sustaining online healthcare services.

Keywords: Group Decision Support Systems, Incomplete Information, Quality-of-Information, e-Health, Multi-valued Extended Logic Programming.

1 Introduction

Defective information is ubiquitous; we take most of our decisions, if not all, based on incomplete, not precise and even uncertain information. Also, a major factor in the flexibility of human reasoning about complex systems comes from the natural ability to use partial information and to combine it according to its availability. Most information systems just ignore this characteristic of the information about the real world and build upon models where some idealisation expunges the inherent uncertainty [1]. The result is a system that never provides the expected answers, due to its inability to model the world. Instead, one should deal with the uncertainty in the model itself. Indeed, to implement useful information systems, namely knowledge based ones, it is necessary to represent and reason with defective information.

Several approaches to the representation of imperfect knowledge may be found in the literature. Many of them link logic with the theory of probabilities, combining Bayesian reasoning, Certainty Factors [2], Dempster-Shafer theory [3], Fuzzy Logic [4, 5], or non standard logics [6]. In the context of a Qualitative Uncertainty Theory, in [7] is presented an axiomatic based on the substitution of the probabilistic interval [0..1] with an ordered set of symbolic values for a multi-valued predicate logic. It approaches the problem of the representation but not the QoI. In another work [8], Dempster-Shafer theory is combined with fuzzy sets in order to represent separately uncertainty and imprecision, for modeling processes of medical diagnoses. Dempster-Shafer is again used in [9] combined with the Analytic Hierarchy Process (AHP) [10] to construct a method for solving Multi-Attribute Decision Making problems with incomplete information.

Following Belnap work [11], Ofer Arieli and Arnon Avron [12] suggest that a four-valued logic is better suited to handle inconsistent or uncertain information than a three-valued logic, for computerized reasoning. They concluded that the existence of elements like $\top$ (lack of information, or "no knowledge") and $\bot$ (inconsistency, or "over-knowledge"), as well as the idea of ordering data according to degrees of knowledge, in a bilattice structure is particularly suitable for reasoning with uncertainty. On the other hand, qualitative models and qualitative reasoning have been around in Artificial Intelligence research for some time [13] [14] [15]. Examples of such systems are GDSS based on agent perception, that in this work will be associated with the healthcare practice and respective information systems (e-Health systems), in which the absence of quantification of the QoI is a blunder [16].

In this work we will focus, therefore, on a new class of systems, such as the VirtualECare [17], which is briefly described below. It represents a new and cost effective way for health care delivery and is in the intersection of Telemedicine, Virtual Healthcare and Electronic Medical Records, leading us to foresee and define a method to evaluate the QoI involved in a GDSS, and to present the foundations of a logical theory that will allow us to represent and reason with defective knowledge. This QoI evaluation method, suitable for hybrid neuro-symbolic algorithms and models for knowledge management [18], is then embedded in a model for group decision making, where qualitative data may be transformed in quantitative measures [19].

Due to the increasing number of different patient management options, combined with the healthcare authorities expectations about the behavior of their physicians, when balancing benefits for the patient against financial costs, there is a growing need to offer support in their decision-making process. Clinical guidelines are an example of this support, and checking whether a guideline conforms to global medical quality requirement is a main issue. Lucas and Hommersom [20] work is a good example of quality evaluation using logic. We use a similar type of knowledge representation, but we extended their approach to consider multi-valued non-monotonic logics in the formalized versions of medical guidelines.

This paper is organized as follows. Section 2 introduces e-Health systems. Section 3 briefly presents the VirtualECare project. In Section 4 is presented the knowledge representation and reasoning system used for the QoI method, presented in Section 5. In Sections 6 and 7 are described the VirtualECare GDSS architecture and Decision Making model, respectively. The main conclusions of this paper are discussed in Section 8.

2 e-Health Systems

One's habitat is rapidly becoming the new place where to center medical and health provision due to the use of Internet technology. People with chronic diseases or conditions are already participating in e-care on a daily basis by entering health data, participating in on-line support groups, downloading information related to their conditions from secured web services and more. Patients are becoming more involved and spend more time on their health care decisions [21]. On the other hand one has to take into account the aging of population, with the associated health problems; it will be impossible to keep all patients in hospitals or other healthcare facilities. The society is faced with the responsibility of taking care of these people in the best possible social and economical ways. Indeed, there is a clear interest to create devices and environments allowing the patients to be followed in their own homes or during their everyday life. Pervasive healthcare and ubiquitous Ambient Intelligence (AmI) home environments are being proposed for this purpose [22] [23] [24].

In the last years we have assisted to a proliferation of various research projects in order to increase the Quality-of-Care (QoC) services and reduce the associated costs, especially those requiring the patient to be delocalized from his home. The following is a very brief description of some projects in this area.

IST Vivago® - This system (IST International Security Technology, Helsinki, Finland) is an active social alarm system, which combines intelligent social alarms with continuous remote monitoring of the user's activity profile [22]. This system is able to provide long term monitoring of the user's circadian rhythm, which, in turn, allows the surveillance of the user wellbeing. The system has been designed chiefly to address the needs of elderly home care and institutional care facilities. The Vivago® system consists of three main components: a wrist unit, a base station and an alarm receiving and routing software. It has two main system configurations: the home one, intended for users living independently in their own home, and the institutional one for user in sheltered or nursing homes.

The Intelligent Butler - The main purpose of this project is to design an intelligent agent, the butler, for aiding disabled and elderly people in a virtual domotics environment [25]. The environment specification has a set of components and devices capable of generating perceptions as well as generating and/or receiving a set of actions. The basic components of the environment are: the domotics house, the user (disabled or elderly person), the daily planner, the user's report and the butler agent. The domotics house, the disabled person and the daily planner are in charge of generating perceptions. The actions are taken by the butler, whose effects are manifested on the domotics house, the disabled person, the daily planner and the report of the user. The environment also comprises several functional modules, such as a healthcare module, related with the health and medical prescriptions for user's treatments, and a nursing module, in charge of planning the daily menu of the user.

Amigo – Ambient Intelligence for the Networked Home Environment - This project is funded by the European Commission as an Integrated Project (IP) in the Sixth Framework Programme, joining together fifteen companies to exploit the potential of the network connectivity of almost every nowadays electronic equipment [26]. The Amigo project develops middleware that dynamically integrates heterogeneous systems to achieve interoperability between services and devices, including home appliances. It ended in February 2008 and has developed an open-source software layer enabling the creation of intelligent networked home environments. It also developed several applications to show the potential for end users, like "Home Care and Safety", "Home Information and Entertainment", and the "Extended Home Environment" in which multiple homes are connected. This project used 'home laboratories' across Europe including Philips Research's 'HomeLab', France Telecom's 'Creative Studio Lab' and the Fraunhofer Institute's 'InHaus'.

I.L.S.A. – The Independent Lifestyle Assistant - This project is an initiative oh the University of Minnesota. It's main objective is to study the response of elderly to a monitoring computer system inside their houses and to determine how such systems can help this handicapped persons [27]. This project not only determined the main problems of elder people living alone, but also implemented parts of a monitoring system in some real conditions test houses. This application of AmI is useful for the elder living alone and also for their caregivers, as everyone maintains its autonomy.

The test case involved eleven houses during half year with a set of sensors dedicated to monitor specific actions (e.g. behavioural patterns, medication taken). The data from the sensors was sent to a central where it was studied. The main features implemented included alerts and notifications, cognitive support (e.g reminders), reports of user behaviour and passive monitoring (e.g. mobility, occupancy, sleeping patterns). Users had a portable device with which they could check their agenda, change some system parameters and even communicate with caregivers. This is a project aimed to a specific public: elder people living alone.

Telecare – This project [28] aims to develop a configurable framework focused on virtual communities for assisting elderly people, based on the integration of multi-agent systems and a federated information management approach. The objective is to provide services useful for elderly people, making use of the emerging ubiquitous computing and intelligent home appliances. The project expects to address issues like elderly people being moved from their homes, providing them with autonomy and independence. To achieve these objectives, this project is based on tele-supervision and tele-assistance technologies. Using existing technologies and the current internet-based infrastructures, this project believes it is possible to create a virtual network, which connects the elderly home to the relative's work or home, to the care or leisure centres and to virtual shops, among others.

All these systems have to deal with information from several sources, sometimes in a distributed processing environment, prone to various kinds of errors, like sensors reading errors, communication noise, misinterpreted data, and generally wrong evaluation of state of affairs. They need to incorporate information about the real world and common sense knowledge, that is to say incomplete, inconsistent, default, and even contradictory, data and knowledge. This is an aspect that is almost ignored in all the aforementioned systems, besides low level error detection and recovery protocols. In conclusion, we may say that, from our point of view, the QoI is not controlled.

3 The VirtualECare Project

The systems presented in the previous chapter suffer from a common characteristic: they tend to be simple and/or basic reactive systems without key requirements from a support platform point of view. In our opinion these systems were very useful to delineate a path for others to follow. Taking this view we present the VirtualECare [29] project which, it is believed, will make the next generation of remote proactive healthcare systems, including, as it is the present case, Group Decision techniques for problem solving.

To better understand the amplitude of VirtualEcare, let's consider the following scenario [30]:

"*John has a heart condition and wears a smart watch that takes his blood pressure three times a day. His watch also reminds him to take his medications and the proper dosage for each medicine. If anything is unusual, his watch alerts both him and the GDSS. John also has a PDA that contains an interactive health control table where he can monitor his medications, schedule his exercises, manage his diet and log his vital statistics. The GDSS has access to this table so they can keep up to date on his condition. Currently, John's watch detects that his blood pressure is unusually high. The GDSS receives a grade B and calls him to check what might be causing his high blood pressure (diagnose). At the same time John receives a checklist of possible causes to review. John compares this list to his own health control table in his PDA to see what might be wrong. Meanwhile, the GDSS decides John should come to an appointment.*"

The presented scenario requires an infrastructure to support all the different actors and to provide basic interaction mechanisms. On top of this infrastructure an extensive number of services may be deployed and/or be developed.

The VirtualECare project embodies an intelligent multi-agent system aimed to monitor and interact with its users, targeted to elderly people and/or their relatives. The system is designed to have several services, beyond the health related ones. It will be connected not only to healthcare institutions, but also with user's relatives, leisure centers, training facilities and shops, just to name a few.

The VirtualECare has a distributed architecture, with components interconnected through different networks (e.g. LAN, MAN, WAN), each one with a specific role. The summarized description, depicted in Figure 1 under a top-level perspective, of the architectural component's roles, is given below.

Supported User's Home – The supported user's natural surroundings, mainly the house where the elderly, or person with special health care needs, lives. The ambient is under constant monitoring by data collecting electronic devices. This data is sent to the **GDSS** through the ***CallCareCenter,*** being of the clinical type, or to the ***CallServiceCenter*** for all the others data types, allowing for a better (Intelligent) Ambient. The user himself/herself has constant vital data supervision and may interact with all services by means of traditional computerized and/or communication devices or other specialized electronic devices.

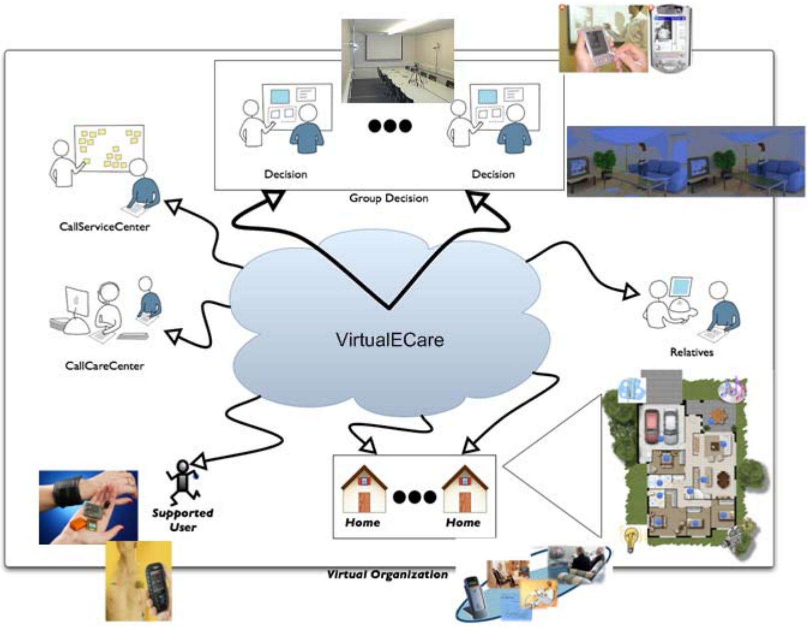

Figure 1. The System Architecture, VirtualECare.

Supported User – Elderly or person with special health care needs, in constant monitoring, allowing collecting of vital data information sent to the ***CallCareCenter*** and forwarded to the ***GDSS***.

Group Decision Support System – There can be more than one group decision, being responsible for the decisions taken in the VirtualECare. The groups are created dynamically according to the problem that needs to be solved. Our work is centered in this key component. An Electronic Health Record of the ***SupportedUser*** exists, allowing a better understanding of his/her special needs and personal preferences.

Call Service Center – Entity with all the necessary computational and qualified personal resources that receives, analyzes the diverse data, and takes the necessary steps according to it.

Call Care Center – Entity with all the necessary computational and qualified personal resources (in this case, healthcare professionals and auxiliary), being capable of receiving and analyzing the healthcare related data and take the necessary actions according to it.

In VirtualECare we identified the same problem about defective information, suffered from the systems presented in chapter 2. It is a problem with more impact in VirtualECare owing to its proactive characteristic. In the following chapters we propose a solution to this problem.

4 Knowledge Representation and Reasoning Systems

The knowledge representation and reasoning systems in a knowledge-driven GDSS is nuclear to the success of the overall operation of such systems [31-33].

In the past few decades, many non-classical techniques for modeling the universe of discourse and the reasoning procedures of intelligent systems have been proposed [34], in addition to classical ones. Of particular interest to this work are the techniques to deal with *incomplete*, *inconsistent*, *contradictory* and *default* information [35].

Intelligent systems require the ability to reason with incomplete information, because in the real world complete information is hard to obtain, even in the most controlled situation. Classic two-valued logics may not be sufficient, chiefly in Multi-Agent Systems (MAS) where an agent may fall in a state of partial ignorance [36].

In addition, an agent has to deal with information coming from different sources or other agents. Then, inconsistencies may arise, because information received by an agent from different agents may not always agree, as happens, for instance, in the medical domain when several experts have different opinions about a given diagnostic.

The idea behind default information is the capability of making assumptions or jump to a plausible conclusion, derived from a knowledge base in the absence of information to the contrary. The derived information is defeasible, because in light of new information the conclusion may need to be retracted, i.e., we are in presence of Non-Monotonic Reasoning [33].

A suitable logic is needed, one that permits the representation of incomplete, inconsistent and default information and supports non-monotonic reasoning.

Historically, uncertain reasoning has been associated with Probability Theory, embodying non-Bayesian theories of subjective probability, as in the Dempster-Shafer Theory [3]. The Dempster-Shafter Theory is well-known for its usefulness to express uncertain judgments of experts. This theory introduces the concept of *belief functions* and is based on two ideas: (i) obtaining degrees of belief for one question from subjective probabilities for a related question, and (ii) Dempster's rule for combining such degrees of belief when they are based on independent items of evidence. However, the use of belief functions may involve challenging computational problems. Beliefs are also represented in other contexts, for example in MAS, where specialized classes are used to model a way of things, propositions or other information relevant to the system and its mental model [37].

Another promising computational paradigm, Abductive Logic Programming (ALP) [38] has been recognized as a way to resolve some limitations of logic programming with respect to higher level knowledge representation and reasoning tasks. Abduction is a way of reasoning on incomplete or uncertain knowledge, in the form of hypothetical reasoning, more appropriate to *model generation* and *satisfiability checking*.

In a classical logical theory, the proof of a question results in a true or false value, or is made in terms of representing something about one could not be conclusive. In opposition, in a logic program, the answer to a question is only of two types: *true* or *false*. This is a consequence of the limitations of the knowledge representation in a logic program, because it is not allowed explicit representation of negative information. Additionally, the operational semantics applies the Closed-World Assumption (CWA) [39] to all the predicates. The generality of logic programs represents implicitly negative information, assuming the application of reasoning according to the CWA.

A logic program is a finite set of clauses in the form:

$$A_0 \leftarrow A_1 \wedge ... \wedge A_m \wedge not\ A_{m+1} \wedge ... \wedge not\ A_n \qquad (1)$$

such as $\forall i \in N_0$, A is an atom and the terms A_i and $not\ A_i$ are literals.

Weak negation - operator *not* in conventional Logic Programming (LP) - is the ***negation-by-failure***: ***not A*** is true if it is not possible to prove ***A***, and ***not A*** is false when is possible to prove ***A***. This kind of reasoning would be enough in a CWA system, but is insufficient when there is incomplete information.

An Extended Logic Program (ELP), on the other hand, is a finite collection of rules of the form [40]:

$$q \leftarrow p_1 \wedge ... \wedge p_m \wedge not\ p_{m+1} \wedge ... \wedge not\ p_{m+n} \qquad (2)$$

$$?\, p_1 \wedge ... \wedge p_m \wedge not\ p_{m+1} \wedge ... \wedge not\ p_{m+n} \qquad (3)$$

where **?** is a domain atom denoting falsity, and q and every p_i are literals, i.e. formulas like a or $\neg a$, being a an atom, for $m, n \in N_0$.

ELP introduces another kind of negation: strong negation, represented by the classical negation sign ¬. In most situations, it is useful to represent ¬*A* as a literal, if it is possible to prove ¬*A*. In EPL, the expressions *A* and *not A*, being *A* a literal, are extended literals, while *A* or ¬*A* are simple literals. Intuitively, *not p* is true whenever there is no reason to believe *p*, whereas ¬*p* requires a proof of the negated literal.

The issue is providing expressive power for representing explicitly negative information, as well as directly describe the CWA for some predicates, also known as *predicate circumscription* [41] [42]. Three types of answers to a given question are then possible: *true*, *false* and *unknown*. The representation of null values will be scoped by the ELP. We consider two types of null values: the first will allow for the representation of unknown values, not necessarily from a given set of values, and the second will represent unknown values from a given set of possible values. In the following, we consider the extensions of the predicates that represent some information about the user home environment, wittingly simple:

```
env_temp: Room x Value
env_humidity: Room x Value
env_lux: Rom x Value
```

The first argument denotes the room and the second represents the value of the property (e.g., `env_temp(bedroom, 20)` means that the environment temperature in the bedroom has the value `20`).

```
(1)  ¬env_temp(E,V)←
            not env_temp(E,V),
            not exception(env_temp(E,V))
(2)  env_temp(bedroom, 20)
```

Program 1 - Knowledge representation in terms of the extension of predicate `env_temp` referred to above, with respect to the bedroom temperature, with a known value

In Program 1, the symbol ¬ represents the strong negation, denoting what should be interpreted as false, and the term *not* designates negation-by-failure.

Now admit that the environment temperature of another room, the *living_room*, has not, yet, been established. This will be denoted by a null value of the type unknown, and represents the situation depicted in Program 2: the room has some environment temperature but it is not possible to be certain (affirmative) about its value. In the second clause the symbol ⊥ represents a null value of an undefined type. It is a representation that assumes any value as a viable solution, but without being given a clue to conclude about which value one is speaking about. It is not possible to compute, from the positive information, the value of the environment temperature for *living_room*. The fourth clause (the closure of predicate `env_temp`) discards the possibility of being assumed as false any question on the specific value of the environment temperature for *living_room*.

```
(1)  env_temp(bedroom, 20)
(2)  env_temp(living_room,⊥)
(3)  ¬env_temp(E,V)←
           not env_temp(E,V),
           not exception(env_temp(E,V))
(4)  exception(env_temp(E,V))←
           env_temp(E,⊥)
```

Program 2 – Knowledge representation in terms of the extension of predicate `env_temp` referred to above, with respect to the temperature of the living room, with an unknown value

In Program 3 the value of the environment temperature for *kitchen* is foreseen to be 20, with a margin of error of 5. It is not possible to be positive, concerning the temperature value. However, it is false that the environment temperature has a value of 14 or 27, for example. This situation suggests that the lack of knowledge may be associated to an enumerated set of possible known values.

As a different case, let us consider the environment temperature of the *dining_room*, that is unknown, but one knows that it is specifically 22 or 25.

```
(1)  env_temp(bedroom, 20)
(2)  env_temp(living_room,⊥)
(3)  ¬ env_temp(E,V)←
           not env_temp(E,V),
           not exception(env_temp(E,V))
(4)  exception(env_temp(E,V))←
           env_temp(E,⊥)
(5)  exception(env_temp(kitchen,V))←
           V ≥ 15 ∧ V ≤ 25
(6)  exception(env_temp(dining_room,22))
(7)  exception(env_temp(dining_room,25))
```

Program 3 – Knowledge representation in terms of the extension of predicate `env_temp` referred to above, with respect to the temperature of the four compartments

Using ELP, as the logic programming language, a procedure given in terms of the extension of a predicate called `demo` is now presented. This predicate allows one to reason about the body of knowledge presented in a particular domain, set on the formalism referred to above. Given a question, it returns a solution based on a set of assumptions. This meta-predicate is defined as `Demo: Question x Answer.`

Where Question indicates a theorem to be proved and Answer denotes a truth value (see Program 4): true (T), false (F), being U in the range of truth values]0…1[.

```
demo(Q,T)← Q
demo(Q,F)← ¬Q
demo(Q,U)← not Q ∧ not ¬Q
```

Program 4. Extension of meta-predicate `demo`

5 Quality-of-Information

Let i ($i \in 1,\ldots, m$) represent the predicates whose extensions make an extended logic program that models the universe of discourse and j ($j \in 1,\ldots, n$) the attributes of those predicates. Let $x_j \in [min_j, max_j]$ be a value for attribute j. To each predicate is also associated a scoring function $V_{ij}[min_j, max_j] \rightarrow 0 \ldots 1$, that gives the score predicate i assigns to a value of attribute j in the range of its acceptable values, i.e., its domain (for simplicity, scores are kept in the interval [0 … 1]), here given in the form:

```
all(attribute_exception_list, sub_expression, invariants)
```

This denotes that *sub_expression* should hold for each combination of the exceptions of the extensions of the predicates that represent the attributes in the *attribute_exception_list* and the *invariants*. The invariants are integrity constraints in the form:

$$\leftarrow p_1 \wedge \ldots \wedge p_m \wedge not\ p_{m+1} \wedge \ldots \wedge not\ p_{m+n} \qquad (4)$$

where all p_i are literals, i.e. formulas like a or $\neg a$, being a an atom, for $m, n \in N_0$.

This is further translated by introducing three new predicates. The first predicate creates a list of all possible exception combinations (pairs, triples, ..., n-tuples) as a list of sets determined by the domain size (and the invariants). The second predicate recurses through this list and makes a call to the third predicate for each exception combination. The third predicate denotes sub_expression, giving for each predicate, as a result, the respective score function. The QoI with respect to a generic predicate P is therefore given by $QoI_P = 1/Card$, where Card denotes the cardinality of the exception set for P, if the exception set is disjoint. If the exception set is not disjoint, the quality of information is given by:

$$QoI_P = \frac{1}{C_1^{Card} + \cdots + C_{Card}^{Card}} \qquad (5)$$

where C_{Card}^{Card} is a card-combination subset, with *Card* elements.

The next element of the model to be considered is the relative importance that a predicate assigns to each of its attributes under observation, i.e. w_{ij} stands for the relevance of attribute *j* for predicate *i*. It is also assumed that the weights of all predicates are normalized, that is:

$$\forall i \sum_{j=1}^{n} w_{ij} = 1 \qquad (6)$$

It is now possible to define a predicate's scoring function, i.e., for a value $x = (x_1, ..., n)$ in the multi dimensional space defined by the attributes domains, which is given in the form:

$$V_i(x) = \sum_{j=1}^{n} w_{ij} * V_{ij}(x_j) \qquad (7)$$

it is viable to measure the QoI that occurs as a result of invoking a logic program to prove a theorem (e.g. `Theorem`), by posting the $V_i(x)$ values into a multi-dimensional space and projecting it onto a two dimensional one.

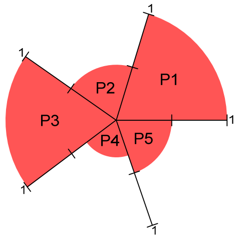

Figure 2 - A measure of the QoI for a logic program or theory P

Using this procedure, it is defined a circle, as the one given in Figure 2. Here, the dashed n-slices of the circle (in this example built on the extensions of five predicates, named as p_1 ... p_5) denote de QoI that is associated with each of the predicate extensions that make the logic program and were used to prove the theorem `Theorem` referred to above.

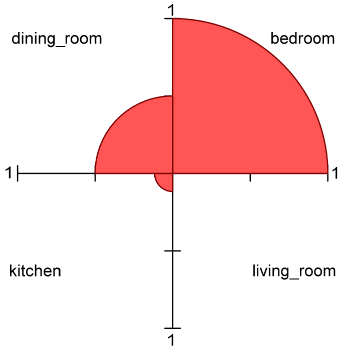

Figure 3 – QoI about the question "Which is the temperature of the user's environment?"

As an example we represent the QoI associated with the information about a user's environment, from Program 3, depicted in Figure 3. In order to find the relationships among the extensions of these predicates, we evaluate the relevance of the QoI, given in the form:

$$\begin{aligned}
&V_{env\_temp}(\texttt{bedroom}) = 1\\
&V_{env\_temp}\ (\texttt{living\_room}) = 0\\
&V_{env\_temp}(\texttt{kitchen}) = 0.1\\
&V_{env\ temp}(\texttt{dining\_room}) = 0.5
\end{aligned} \qquad (8)$$

It is now possible to measure the QoI associated to the question put in the context of the logic program env_temp referred to above, which was given in the form "Which is the temperature of the user environment?". The shaded n-slices (here n is equal to four) of the circle denote the QoI.

6 Group Decision Support Systems

Collaborative work has inherent advantages (e.g., better knowledge, different world perspectives, increased acceptance) but also presents some drawbacks (e.g., social pressure, domination, goal displacement, group thinking) [43].

GDSS have been the subject of much research, have matured over a period of many years and there are many instances of their successful application [44] [45]. One of the main drawbacks of many GDSS implementations is the Problem-Processing System (PPS) [45], overloaded to provide strong coordination for handling or even guiding partakers interactions, linked with the abilities of knowledge acquisition, and to incorporate this knowledge into the Knowledge System (KS).

The VirtualECare GDSS is a knowledge-driven Decision Support Systems (DSS) [46], that relies on database and model representations of the world, following a proof-theoretical approach to computing that addresses the truth value of a question (or theorem to be proved) in terms of the QoI of the terms that make the extension of a predicate or predicates under invocation [47].

The VirtualECare GDSS (Figure 4) has a traditional architecture. The **User Interface** module incorporates a **Language System** (all messages the GDSS can accept) and a **Presentation System** (all messages the GDSS can emit). The **Data Management**, **Model Management** and **Knowledge Management** modules, along with the respective representations, make up the overall **Knowledge System**, i.e., all the knowledge de GDSS has stored and retained. The **Problem Processing** module is the GDSS software engine, the active component of the system. It can be triggered by events that are detected outside or inside the system. The **Problem Processing** module incorporates the ability to evaluate the QofI available in any decision process.

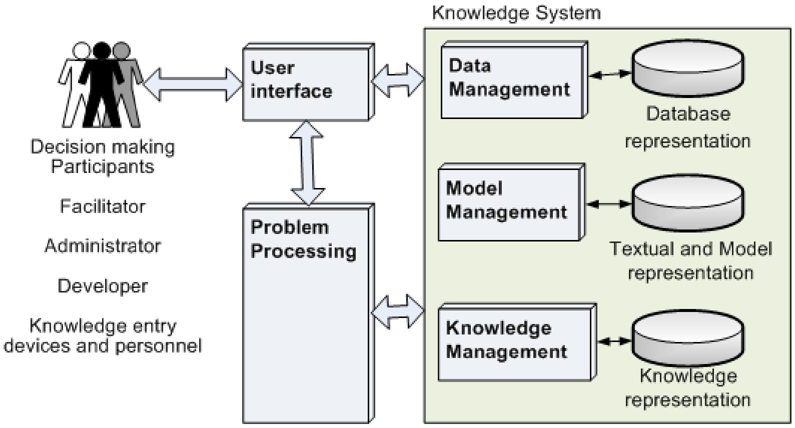

Figure 4 – Top-level architecture of VirtualECare GDSS

7 Decision Making in VirtualECare

The staged nature of decision making processes is established by several studies [48, 49]. The VirtualECare GDSS follows this procedural staged nature, as it is depicted in Figure 5. We named *meeting* to all the phases necessary to the completion of a specific task, i.e., a meeting happens as a result of the interaction between two or more individuals [50]. A meeting can be realized in one of the four scenarios: i) same time / same place; ii) same time / different places; iii) different times / same place; iv) different times / different places. Each one of these scenarios will compel from the GDSS a different kind of action.

The underlying process model of the VirtualECare GDSS follows Simon's empirical rationality [51]. The *Intelligence* stage occurs continuously, as the GDSS interacts with other components of the VirtualECare system. Identified problems that call for an action triggers the formation of a group decision. This group formation is conducted in the pre-meeting phase, when a facilitator must choose the partakers. In order to form the "best" group he/she must evaluate the QoI on hand of possible participants, and not about the participants themselves, registered in the knowledge system. The *Design* and *Choice* phases occur in the in-meeting stage. In the In-Meeting phase, the participants will be working in order to accomplish the meeting goals and to take de finest decisions. In order to accomplish this goal, the participants use a knowledge database and exchange information. Once again, the system must provide a measure of the QoI available. In the Post-Meeting phase it is important to evaluate the results achieved so far by the group, as well as how much each group member is acquit with the achieved results (satisfied/unsatisfied).

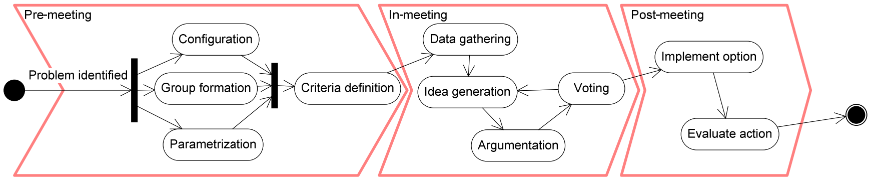

Figure 5 – The staged nature of a decision process model

The in-meeting stage cycles through a set of iterations, similarly to the *circular logic of choice* of Nappelbaum [52]. In Nappelbaum model a sharpening spiral of the description of the problem cycles through option descriptions, value judgments and instrumental instructions, towards a prescribed choice. We further extend this approach, in line with Jones and Humphreys model of the Decision Hedgehog [48]. Instead of constructing and prescribing the solution to the decision problem within a procedural context of a single decision path, we suggest the exploration of potential different pathways to develop contextual knowledge, enabling collaborative authoring of outcomes.

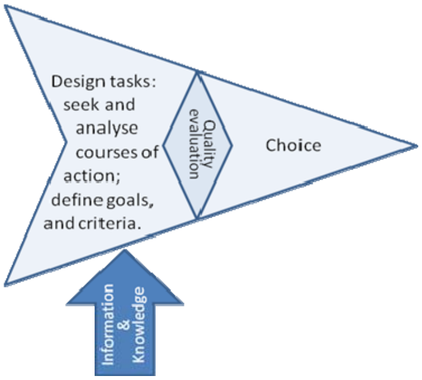
Design tasks: seek and analyse courses of action; define goals, and criteria.
Quality evaluation
Choice
Information & Knowledge

Figure 6 – In-meeting stage: design and choice phases separated by the evaluation of the QofI

In this way, the QoI is evaluated within each iteration, for every possible pathway. The knowledge system is scanned for the needed information with a previously agreed threshold of the QoI being measured [53] [54]. If the QoI measure does not reach the necessary threshold, new information and/or knowledge is searched for and the process restarts. Figure 6 illustrates this process for a single iteration, while Figure 7 depicts the situation where the quality threshold is only reached on a step-by-step process, attaining a point in time when the decision is made. In each iteration, we can use different approaches to generate alternatives and criteria, namely Idea Generation, Argumentation and Voting techniques, to support the decision-making process [53].

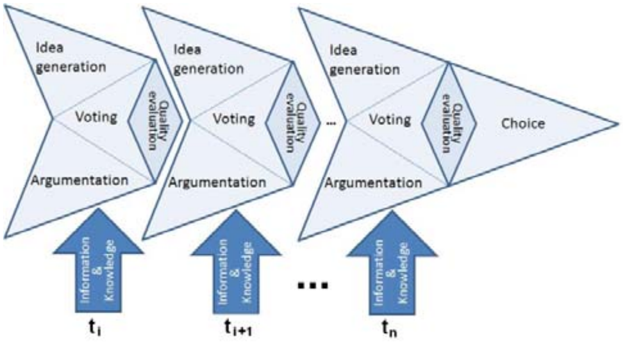

Figure 7 – In-meeting stage with several iterations

Even when time compels the group to make a decision, well before the quality threshold had been reached, the evaluation of the QoI involved in the inference process is of great utility, once it gives us a measure of the confidence that we must put on the decision itself.

8 Conclusions and Future Work

It was presented a model for group decision making where the evaluation of QoI of the partakers involved plays a central role. We use an Extended Logic Programming language for knowledge representation and reasoning with defective information, allowing one to model the universe of discourse in terms of *incomplete*, *inconsistent*, *contradictory* and *default* data and knowledge. We also present an architecture of a Group Decision Support System in the context of the VirtualECare project, a system aimed at sustaining online healthcare services, that embodies all the functionalities referred to above.

The aforementioned environment configures a virtual organization where the information is collected from several sources in a time/space flexible way, being sensor data from devices, information from non-experts (e.g., the user relatives) or knowledge from medical experts. This diversity of sources and time/space flexibility poses special problems for evaluation of the QoI. In this context, the perception of the QoI from all the participating actors is a critical factor to accomplish a high degree of confidence in the information flowing in the systems, in the other participants and in the system as a whole. The quantification of the QoI is by itself another decision criteria against each alternative may be evaluated.

It was made a systematization of the evolution of the body of knowledge about QoI embedded in decision-making process models. A proposal that tries to solve the representation problem of defective information was presented, adequate for evaluation of QoI in such a process. It was also presented a computationally feasible formal tool to measure the value of QoI.

These proposals were applied to decision-making in the field of healthcare systems, in the context of VirtualECare, with an underlying framework of Clinical Guidelines.

Using the work presented here as a foundation, we can foresee future developments, such as the system being able to make recommendations on how to progress in the decision making, using Case Based Reasoning, under an evolutionary intelligence based approach.

References

1. Parsons, S., *Current approaches to handling imperfect information in data and knowledge bases.* IEEE Trans. on Knowledge and Data Eng., 1996. **8**(3): p. 353-372.
2. Buchanan, B.G. and E.H. Shortliffe, *Rule Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project (The Addison-Wesley series in artificial intelligence).* 1984: Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.
3. Shafer, G., *The Dempster-Shafer theory*, in *Encyclopedia of Artificial Intelligence, 2nd Ed*, S.C. Shapiro, Editor. 1992, Wiley.
4. Zadeh, L.A., *Fuzzy logic.* Scholarpedia, 2008. **3**: p. 1766.
5. Haslum, K., A. Abraham, and S. Knapskog. *HiNFRA: Hierarchical Neuro-Fuzzy Learning for Online Risk Assessment.* in *Second Asia International Conference on Modeling and Simulation, AMS*. 2008.
6. Caicedo, X., F. Dechesne, and T.M.V. Janssen, *Equivalence and quantifier rules for logic with imperfect information.* Logic Jnl IGPL, 2009. **17**(1): p. 91-129.
7. Seridi, H. and H. Akdag, *A qualitative approach for processing uncertainty.* Uncertainty in intelligent and information systems, 2000. **20**: p. 46–57.
8. Straszecka, E., *Combining uncertainty and imprecision in models of medical diagnosis.* Information Sciences, 2006. **176**(20): p. 3026-3059.
9. Hua, Z., B. Gong, and X. Xu, *A DS-AHP approach for multi-attribute decision making problem with incomplete information.* Expert Systems with Applications, 2008. **34**(3): p. 2221-2227.
10. Saaty, T.L., *The Analytic Network Process*. 2001: RWS Publications.
11. Belnap, N.D., *How a computer should think.* Contemporary aspects of philosophy, 1977: p. 30–56-30–56.
12. Arieli, O. and A. Avron, *The value of the four values.* Artificial Intelligence, 1998. **102**(1): p. 97-141.
13. Weld, D.S. and J. De Kleer, *Readings in qualitative reasoning about physical systems.* 1989: Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.
14. Kuipers, B., *Qualitative reasoning: modeling and simulation with incomplete knowledge*. 1994: MIT press.
15. Forbus, K.D., *Qualitative reasoning.* CRC Handbook of Computer Science and Engineering, 1996: p. 715-733.
16. Cruz-Correia, R.J., P.M. Vieira-Marques, A.M. Ferreira, F.C. Almeida, J.C. Wyatt, and A.M. Costa-Pereira, *Reviewing the integration of patient data: how systems are evolving in practice to meet patient needs.* BMC Medical Informatics and Decision Making, 2007. **7**(14).
17. Costa, R., J. Neves, P. Novais, J. Machado, L. Lima, and C. Alberto, eds. *Intelligent Mixed Reality for the Creation of Ambient Assisted Living*. Progress in Artificial Intelligence, ed. J. Neves, M. Santos, and J. Machado. Vol. 4874. 2007, Lecture Notes in Artificial Intelligence - Spinger.
18. Herrero, Á., E. Corchado, L. Sáiz, and A. Abraham, *DIPKIP: A CONNECTIONIST KNOWLEDGE MANAGEMENT SYSTEM TO IDENTIFY KNOWLEDGE DEFICITS IN PRACTICAL CASES.* Computational Intelligence, 2010. **26**(1): p. 26-56.

19. Baruque, B. and E. Corchado, *A weighted voting summarization of SOM ensembles.* Data Mining and Knowledge Discovery, 2010.
20. Hommersom, A., P.J.F. Lucas, and P. van Bommel, *Checking the quality of clinical guidelines using automated reasoning tools.* Theory and Practice of Logic Programming, 2008. **8**(5-6): p. 611-641.
21. Goldstein, D.E., *E-healthcare: Harness the Power of Internet E-commerce & E-care*. 2000: Jones & Bartlett. 491.
22. Sarela, A., I. Korhonen, J. Lotjonen, M. Sola, and M. Myllymaki. *IST Vivago® - an intelligent social and remote wellness monitoring system for the elderly*. in *4th International IEEE EMBS Special Topic Conference*. 2003.
23. Ramos, C., *Ambient Intelligence – A State of the Art from Artificial Intelligence Perspective*, in *Progress in Artificial Intelligence*. 2007. p. 285-295.
24. Vallee, M., F. Ramparany, and L. Vercouter. *A multi-agent system for dynamic service composition in ambient intelligence environments*. in *Advances in Pervasive Computing, (Pervasive 2005)*. 2005.
25. Fiol-Roig, G., D. Arellano, F. Perales, P. Bassa, and M. Zanlongo, *The Intelligent Butler: A Virtual Agent for Disabled and Elderly People Assistance*, in *DCAI 2008*. 2009. p. 375-384.
26. IST-004182-Amigo-IP, ***Amigo - Ambient Intelligence for the networked home environment*** 2004, Information Society Technologies.
27. Haigh, K.Z., L.M. Kiff, J. Myers, V. Guralnik, C.W. Geib, J. Phelps, and T. Wagner. *The Independent Lifestyle Assistant tm (ILSA): AI lessons learned*. in *IAAI-04*. 2004.
28. Camarinha-Matos, L.M. and H. Afsarmanesh, *A multi-agent based infrastructure to support virtual communities in elderly care.* IJNVO, 2004. **2**(3): p. 246-266.
29. Costa, R., P. Novais, G. Marreiros, C. Ramos, and J. Neves, *VirtualECare: Group Decision Supported by Idea Generation and Argumentation*, in *PRO-VE 2008*. 2008: Poznan, Poland.
30. Holmlid, S. and A. Björklind, *Ambient Intelligence to Go*, in *AmIGo White Paper on mobile intelligent ambience*. 2003.
31. Way, E.C., *Knowledge Representation and Metaphor*. 1991, Dordrecht, Holland: Kluwer Academic Publishers.
32. Analide, C., P. Novais, J. Machado, and J. Neves, *Quality of Knowledge in Virtual Entities*, in *Encyclopedia of Communities of Practice in Information and Knowledge Management*. 2006, Elayne Coakes and Stev Clarke (Eds).
33. Ginsberg, M.L., *Readings in Nonmonotonic Reasoning*. 1991: Morgan Kauffman Publishers, Inc.
34. Sheridan, F.K.J., *A survey of techniques for inference under uncertainty.* AI Review, 1991. **5**(1): p. 89.
35. Sim, K., *Bilattices and Reasoning in Artificial Intelligence: Concepts and Foundations.* AI Review, 2001. **15**(3): p. 219.
36. Koutras, C.D., C. Nomikos, and P. Peppas, *On a Simple 3-valued Modal Language and a 3-valued Logic of 'not-fully-justified' Belief.* Logic Jnl IGPL, 2008. **16**(6): p. 591-604.
37. Cervenka, R. and I. Trencansky, *The Agent Modeling Language - AML*. 2007: Birkhäuser Verlag AG.
38. Denecker, M. and A. Kakas, *Abduction in logic programming*, in *Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski, Part I*, A. Kakas and F. Sadri, Editors. 2002, Springer. p. 402-436.
39. Hustadt, U. *Do we need the closed-world assumption in knowledge representation?* in *KI'94 Workshop*. 1994. Saarbrüken, Germany: Baader, Buchheit, Nutt (eds.).
40. Gelfond, M. and V. Lifschitz, *Logic Programs with Classical Negation*, in *Logic Programming: Proceedings of the 7th International Conference*, D. Warren and P. Szeredi, Editors. 1990. p. 579-597.
41. Neves, J. *A Logic Interpreter to Handle Time and Negation in Logic Data Bases*. in *Proceedings of the ACM'84, The Fifth Generation Challenge*. 1984.
42. Apt, K.R. and R. Bol, *Logic Programming and Negation: A Survey*. Journal of Logic Programming, 1994. **19**: p. 9-71.
43. Marreiros, G., R. Santos, C. Ramos, J. Neves, P. Novais, J. Machado, and J. Bulas-Cruz. *Ambient Intelligence in Emotion Based Ubiquitous Decision Making*. in *IJCAI 2007 - AITAmI'07*. 2007.
44. Power, D.J., *Decision Support Systems: Concepts and Resources for Managers*. 2002: Greenwood Publishing Group.
45. Burstein, F. and C.W. Holsapple, eds. *Handbook on Decision Support Systems 1: Basic Themes*. International Handbooks on Information Systems. 2008, Springer.
46. Lu, J., G. Zhang, D. Ruan, and F. Wu, *Multi-objective Group Decision Making: Methods, Software and Applications with Fuzzy Set Techniques*. Series in Electric and Computer Engineering. 2007: Imperial College Press.
47. Analide, C., A. Abelha, J. Machado, and J. Neves. *An Agent Based Approach to the Selection Dilemma inCBR*. in *IDC 2008, Catania, Italy, 2008*. 2008.
48. Jones, G. and P. Humphreys. *The Decision Hedgehog: Enhancing Contextual Knowledge for Group Decision Authoring and Communication Support* in *Fifth International and Interdisciplinary Conference on Modeling and Using Context*. 2005. Paris (France): CEUR-WS.
49. Kolbin, V.V., *Decision Making and Programming*. 2003: World Scientific.
50. Bostrom, R., R. Anson, and V. Clawson, eds. *Group facilitation and group support systems*. Group Support Systems: New Perspectives, ed. Jessup and Valachic. 2003, Macmillan.
51. Simon, H.A., *Models of Bounded Rationality: Empirically Grounded Economic Reason*. Vol. 3. 1982: MIT Press.
52. Nappelbaum, E., *Systems logic for problem formulation and choice*, in *IFIP TC8 WG8.3*, P. Humphreys, et al., Editors. 1997, Springer.
53. Lima, L., P. Novais, and J. Cruz, *A Process Model for Group Decision Making with Quality Evaluation*, in *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*. 2009. p. 566-573.
54. Lima, L., R. Costa, P. Novais, C. Analide, J. Bulas-Cruz, and J. Neves, *Quality of Information in the context of Ambient Assisted Living*, in *Advances in Soft Computing*. 2008, Springer- Verlag. p. 624-633.