

in Industrial and Environmental Applications, 5th International Workshop (SOCO 2010), Corchado E., Novais P., Analide C., Sedano J., (Eds.) Springer - Series Advances in Intelligent and Soft Computing, vol. 73, ISBN 978-3-642-13160-8, pp 187-195, (International Workshop on Soft Computing Models in Industrial Applications, Guimarães, Portugal, 16-18th June/2010), 2010.

Decision Making and Quality-of-Information

Paulo Novais¹, Maria Salazar², Jorge Ribeiro³, Cesar Analide¹ and José Neves¹

¹ CCTC, Department of Informatics, University of Minho, Braga, Portugal, email: {pjon, analide, jneves}@di.uminho.pt

² Centro Hospitalar do Porto, EPE, Porto, Portugal, email: msalazar@chporto.min-saude.pt

³ School of Technology and Management, Viana do Castelo Polytechnic Institute, Viana do Castelo, Portugal, email: jribeiro@estg.ipv.pt

Abstract In Group Decision Making based on argumentation, decisions are made considering the diverse points of view of the different partakers in order to decide which course of action a group should follow. However, knowledge and belief are normally incomplete, contradictory, or error sensitive, being desirable to use formal tools to deal with the problems that arise from the use of uncertain and even not precise information. On the other hand, qualitative models and qualitative reasoning have been around in Artificial Intelligence research for some time, in particular due the growing need to offer support in decision-making processes, a problem that in this work will be addressed in terms of an extension to the logic programming language and based on an evaluation of the Quality-of-Information (QoI) that stems out from those extended logic programs or theories. We present a computational model to address the problem of decision making, in terms of a multitude of scenarios, also defined as logic programs or theories, where the more appropriate ones stand for the higher QoIs values.

1. Introduction

Commonly, knowledge and belief are incomplete, contradictory, or error sensitive, being desirable to use formal tools to deal with these problems [1,2]. Logic and Logic programs [3] have emerged as attractive knowledge representation formalism and an approach to solving search problems. In the past few decades, many non-classical techniques for modeling the universe of discourse and reasoning procedures of intelligent systems have been proposed [4, 5, 6]. A part from the need to treat the problem of uncertain information there exists a second need related to the problem of incomplete information. Logic Programming presents a powerful and attractive knowledge representation and reasoning formalism to solve search problems in environments with defective information. For example, Hommersom and Colleagues [7] work is a good example of quality evaluation using logic. They used abduction and temporal logic for quality-

checking of medical guidelines, proposing a method to diagnose potential problems in a timeline, regarding the fulfillment of general medical quality criteria at a meta-level characterization. They explored an approach which uses a relational translation to map the temporal logic formulas to first-order logic and a resolution-based theorem prover.

The objective is to build a quantification process of the Quality-of-Information (QoI) that stems from a logic program or theory during an evolutive process that aims to solve a problem in environments with incomplete information. It is presented a model for group decision making with quality evaluation, along with the several stages of the decision making process in the context of a Group Decision Support System (GDSS) for VirtualECare [9].

2. The Computational Model

With respect to the computational model it was considered an extension to the language of Logic Programming with two kinds of negation, classical negation, \neg , and default negation, *not*. Intuitively, *not* p is *true* whenever there is no reason to believe p (close world assumption), whereas $\neg p$ requires a proof of the negated literal. An Extended Logic Program (ELP for short) [10], on the other hand, is a finite collection of rules of the form [4]:

$$q \leftarrow p_1 \wedge \dots \wedge p_m \wedge \text{not } p_{m+1} \wedge \dots \wedge \text{not } p_{m+n}$$

$$?p_1 \wedge \dots \wedge p_m \wedge \text{not } p_{m+1} \wedge \dots \wedge \text{not } p_{m+n}$$

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 \neg . In most situations, it is useful to represent $\neg A$ as a literal, if it is possible to prove $\neg A$. In EPL, the expressions A and *not* A , being A a literal, are extended literals, while A or $\neg A$ are simple literals.

Every program is associated with a set of abducibles. Abducibles can be seen as hypotheses that provide possible solutions or explanations of given queries, being given here in the form of exceptions to the extensions of the predicates that make the program. To reason about the body of knowledge presented in a particular program or theory, set on the base of the formalism referred to above, let us consider a procedure given in terms of the extension of a predicate called *demo*, using ELP. This predicate allows 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 (*demo*) will be defined as: A meta theorem-solver for incomplete information represented by the signature $\text{demo}: T, V \rightarrow \{\text{true}, \text{false}\}$, infers the

valuation V of a theorem T in terms of the truth values *false* (or 0), *true* (or 1) and *unknown* (with truth values in the interval $]0,1[$), according to the following set of productions:

```
demo(T, true) ← T.
demo(T, false) ← ¬ T.
demo(T, unknown) ← not T, not ¬ T.
```

As a simple example, let us consider the following set of predicates, that stand for themselves:

```
itch: Name x Value
fever: Name x Value
pain: Name x Value
```

where the first argument denotes the name of the patient and the second one the truth value (or degree of confidence) that one has on the former. The extension of predicate *itch* may now be given in the form (program 1):

```
¬itch(X,Y) ← not itch(X,Y), not abducibleitch(X,Y).
abducibleitch(X,Y) ← itch(X,⊥).
itch(kevin,⊥).
itch(john,1).
abducibleitch(carol,0.6).
abducibleitch(carol,0.8).
? ((abducibleitch(X1,Y1) ∨ abducibleitch(X2,Y2)) ∧ ¬
(abducibleitch(X1,Y1) ∧ abducibleitch(X2,Y2)))
```

Program 1. Extension of the predicate *itch*.

where the first clause denotes the closure of the predicate *itch*. In the second clause the symbol ' \perp ' stands for a null value, in the sense that it subsumes that Y may take any truth value in the interval $[0,1]$. The fourth clause denotes that the truth value of *itch* for the patient *john* is 1 . The clauses five and six denote the fact that the truth value of *itch* for patient *carol* is either 0.6 or 0.8 , or even both. The seventh clause stands for the invariant that implements the *XOR* operator, i.e. it states that the truth value of *itch* for the patient *carol* is either 0.6 or 0.8 , but not both.

The extension of predicate *fever* may now be given in the form (program 2):

```
¬fever(X,Y) ← not fever(X,Y), not abduciblefever(X,Y).
```

```

abduciblefever(X,Y) ← fever(X, ⊥) .

fever(carol, ⊥) .

fever(kevin, 1) .

abduciblefever(john, 0.50) .

abduciblefever(john, 0.75) .

?((abduciblefever(X1, Y1) ∨ abduciblefever(X2, Y2)) ∧ ¬
(abduciblefever(X1, Y1) ∧ abduciblefever(X2, Y2))

```

Program 2. Extension of the predicate *fever*

The extension of predicate *pain* may now be given in the form (program 3):

```

¬pain(X,Y) ← not pain(X,Y), not abduciblepain(X,Y) .

abduciblepain(X,Y) ← pain(X, ⊥) .

pain(carol, ⊥) .

pain(kevin, 1) .

abduciblepain(john, 0.3) .

abduciblepain(john, 0.45) .

abduciblepain(john, 0.57) .

```

Program 3. Extension of the predicate *pain*.

In program 3 the last three clauses denote the case where the truth value for *pain* for patient *john* is unknown, although in the set $\{0.3, 0.45, 0.57\}$.

3. Quality-of-Information

In decision making processes [9, 11] it is necessary to search only the most promising search paths. Each path must be tested on their ability to adapt to changing environments, to make deductions and draw inferences, and to choose the most appropriate course of action from a wide range of alternatives. The optimal path in an ELP context is the logic program or theory that models the universe of discourse and maximizes its Quality-of-Information (QoI) factor. Let i ($i \in [1, m]$) represent the predicates whose extensions make an extended logic program that models the universe of discourse, as it is given above in terms of the predicates *itch*, *fever*, and *pain*, where j ($j \in [1, n]$) denote 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 $V_{ij}[min_j, max_j] \rightarrow [0, 1]$, that gives the score predicate i assigned 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 $[0, 1]$).

The *Quality-of-Information* (QoI) with respect to a generic predicate P can be analyzed in four situations and can be measure in the interval $[0, 1]$, when the information is positive and negative, when the information is unknown but can be selected from one or more values, and when the information is unknown but can be derived from a set of values, but only one can be selected. If the information is known (positive) or false (negative) the (QoI) for the predicate term under consideration is 1. For situations where the value is *unknown* the QoI is given by:

$$QoI_P = \lim_{N \rightarrow \infty} \frac{1}{N} = 0 (N \gg 0)$$

For situations when the information is unknown but can be derived from a set of values, $QoI_P = 1/Card$, 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:

$$QoI_P = \frac{1}{C_1^{Card} + \dots + C_{Card}^{Card}}$$

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:

$$\forall i \sum_{j=1}^n w_{ij} = 1, \text{ for all } i.$$

On the another hand, the predicate scoring function, when associated to a value $x=(x_1, \dots, x_n)$ in a multi-dimensional space, it is defined in terms of its attribute domains in the form:

$$V^i(x) = \sum_{j=1}^n w_j^i * V_j^i(x_j)$$

Therefore, 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. For example, for patient john, a logic program or theory P may be got in terms of the logic programs 1, 2 and 3 referred to above, being depicted in the form:

$\neg itch(X,Y) \leftarrow not\ itch(X,Y), not\ abducible_{itch}(X,Y).$

$itch(john,1).$

$\neg fever(X,Y) \leftarrow not\ fever(X,Y), not\ abducible_{fever}(X,Y).$

$abducible_{fever}(john,0.50).$

```

abduciblefever(john,0.75).
?((abduciblefever(X1,Y1) ∨ abduciblefever(X2,Y2)) ∧ ¬
(abduciblefever(X1,Y1) ∧ abduciblefever(X2,Y2))

¬pain(X,Y) ← not pain(X,Y), not abduciblepain(X,Y).
abduciblepain(john,0.3).
abduciblepain(john,0.45).
abduciblepain(john,0.57).

```

whose *QoI* is presented in Figure 1

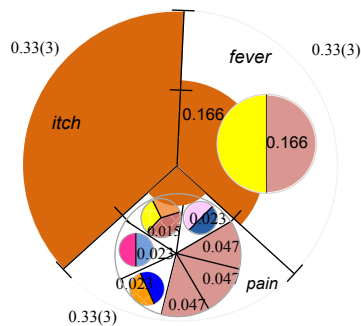


Fig. 1. A measure of the *QoI* for the Logic Program or theory *P* referred to above

4. Decision Making in VirtualECare

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 [9].

The VirtualECare GDSS is a knowledge-driven Decision Support Systems (DSS) [8], that relies on a database (or knowledge base), and models representations of the world, following a proof-theoretical approach to computing, that addresses the truth value of a theorem to be proven in terms of the *QoI* of the terms that make the extension of a predicate or predicates under invocation [10].

Our approach of a VirtualECare GDSS follows Simon's empirical rationality [12]. 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 one must evaluate the *QoI* on hand of possible participants, and not about the participants themselves, registered in the knowledge base 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).

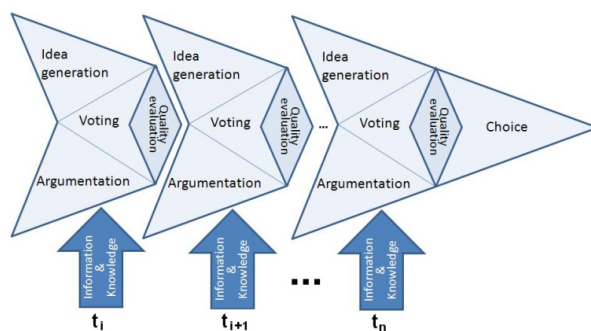


Fig. 2. In-meeting stage with several iterations

The in-meeting stage cycles through a set of iterations, similarly to the *circular logic of choice* of Nappelbaum [13]. 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 [14]. 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.

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 [10, 15]. If the *QoI* measure does not reach the necessary threshold, new information and/or knowledge is searched for and the process restarts. Figure 2 illustrates 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 [8].

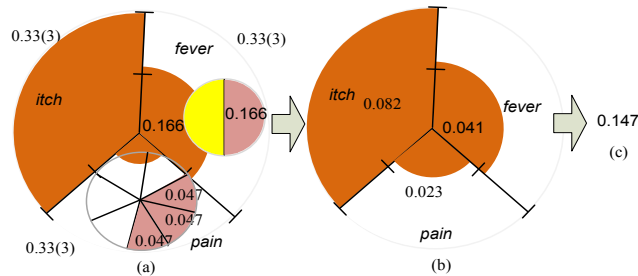


Fig. 3. A measure of the *QoI* of a possible making decision scenario for the patient *john*

Even when time compels the group to make a decision, well before the quality threshold had been reached, the evaluation of the *QoI* that drives the inference process is paramount, once it gives us a measure of the confidence that we put on the decision itself. At any moment, we are faced with different making scenarios, each one with its proper *QoI* (Figure 1). For example, in Figure 3 (a) it is depicted the *QoI* for one scenario that evolve from programs 1, 2 and 3 with respect to the patient *john*, and in (b) the same representation but having now into consideration the predicates relevance. It is now possible to define an order relation over the different scenarios, in terms of its *QoI*; i.e the scenario (theory) to be selected is the one that presents the greatest truth value.

5. Conclusions

Qualitative models and qualitative reasoning have been around in Artificial Intelligence research for some time, in particular due the growing need to offer support in decision-making processes. This area brings together research and evaluation projects in which healthcare decision-making plays a vital role. Indeed, decision-making with healthcare implications spans a broad area, and is relevant at the national, regional, local, and patient levels, and in the public and private spheres. The main focus of our studies in this field is decision-making at the patient level; although, as future work, we intend to study the problem at the organisational and societal levels. Our work addressed the problem of group decision making, modeling it in terms of a multitude of scenarios, defined as logic programs or theories, being its selection based on its soundness, here measured in terms of their *QoI*s values.

References

1. Kowalski, R.: The logical way to be artificially intelligent. In: Toni, F., Torroni, O. (eds), Proceedings of the CLIMA VI.LNCS (LNAI), pp.1-22, Springer, Heidelberg, 2006
2. Sheridan, F.: A Survey of techniques for inference under uncertainty. *Artificial Intelligent Review* 5(1):89, 1991
3. Ginsberg, M. L.: *Readings in Nonmonotonic Reasoning*. Los Altos, California, EUA, Morgan Kaufman Publishers, Inc , 1991
4. Neves J.: A Logic Interpreter to Handle Time and Negation in Logic Data Bases, in Proceedings of ACM'84, The Fifth Generation Challenge, pp. 50-54, 1984
5. Subrahmanian, V.: Probabilistic databases and logic programming, In Proc. of the 17th International Conference of Logic Programming, 2001
6. Kakas, A., Kowalski, R. and Toni, F.: The role of abduction in logic programming. *Handbook of Logic in Artificial Intelligence and Logic Programming*, vol.5, pp.235-324, Oxford U.P., 1998
7. 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* 8(5-6): 611-641, 2008
8. Lima L., Novais P., Bulas Cruz J., A Process Model For Group Decision Making With Quality Evaluation, in *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing and Ambient Assisted Living*, Omatiu S., et al. (Eds.) LNCS 5518, Springer-Verlag, ISBN 978-3-642-02480-1, pp. 566-573, 2009
9. Novais P., Costa R., Carneiro D., Machado J., Lima L., Neves J., Group Support in Collaborative Networks Organizations for Ambient Assisted Living, in *Towards Sustainable Society on Ubiquitous Networks*, Makoto Oya, Ryuya Uda, Chizuko Yasunobu (eds), Springer-Verlag, Series: IFIP, ISBN 978-0-387-85690-2, pp 353-362, 2008
10. Analide, C., Novais P., Machado, M., Neves, J.: Quality of Knowledge in Virtual Entities. In: *Encyclopedia of Communities of Practice in Information and Knowledge Management*, pages: 436-442. Idea Group Inc, USA, 2006
11. 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. Imperial College Press, 2007
12. Simon, H.A., *Models of Bounded Rationality: Empirically Grounded Economic Reason*. Vol. 3. MIT Press, 1982
13. Nappelbaum, E., *Systems logic for problem formulation and choice*, in IFIP TC8 WG8.3, P. Humphreys, et al., Editors, Springer, 1997
14. 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*. Paris (France): CEUR-WS, 2005
15. Machado J., Abelha A., Novais P., Neves J., Neves J., Quality of service in healthcare units, *Int. J. Computer Aided Engineering and Technology*, Vol. 2, No. 4, pp. 436-449, 2010