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, (Proceedings of the International Symposium on Distributed Computing and Artificial Intelligence (DCAI 2009), Salamanca, Spain, 2009), 2009.

A Process Model for Group Decision Making with Quality Evaluation

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

[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; jcruz@utad.pt

Abstract. In this work it is addressed the problem of information evaluation and decision making process in Group Decision Support Systems (GDSS). A Multi-valued Extended Logic Programming language is used for imperfect information representation and reasoning. A model embodying the quality evaluation of the information, along the several stages of the decision making process, is presented. This way we give the decision makers a measure of the value of the information that supports the decision itself. This model is presented in the context of a GDSS for VirtualECare, a system aimed at sustaining online healthcare services. Reasoning with incomplete and uncertain knowledge has to be dealt with in this kind of environment, due to the particular nature of the healthcare services, where the awful consequences of bad decisions, or lack of timely ones, demand for a responsible answer.

Keywords: Group decision support systems, Process model, Quality evaluation

1 Introduction

One of the components of VirtualECare [1] is a knowledge-based GDSS. In this paper we define the architecture of such a GDSS and present a process model that permits to reason with uncertain knowledge. The critical factor that affects de decision making process in contexts similar to VirtualECare is this uncertainty, consequence of the imperfect information about the real world [2]. Several methods for representing and reasoning with imperfect information have been studied [2-5]. We present a method to evaluate the quality of information, in presence of imperfect information, and to control the decision making process itself. The decision must be made only when the quality of information reaches a given threshold or, if the group is compelled by time, at least the participants know the knowledge conditions it was made.

In this paper, we start by briefly presenting the overall architecture of the GDSS, the representation of imperfect information and the method to evaluate its quality. In section 4 we elaborate about the decision process model embodied in the GDSS and how to control the decision progress. Finally, in section 5 we draw some conclusions and future work.

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, (Proceedings of the International Symposium on Distributed Computing and Artificial Intelligence (DCAI 2009), Salamanca, Spain, 2009), 2009.

2 The VirtualECare Project

The VirtualECare project [1] embodies an intelligent multi-agent system aimed to monitor and interact with its users, targeted to elderly people and/or their relatives. These systems will be used to interconnect healthcare institutions, training facilities and leisure centres, shops and patients relatives, on a common network, i.e., the VirtualECare architecture stands for a distributed one with its different nodes answering for a different role, either in terms of a call centre, a group decision support system or a monitoring device, just to name a few.

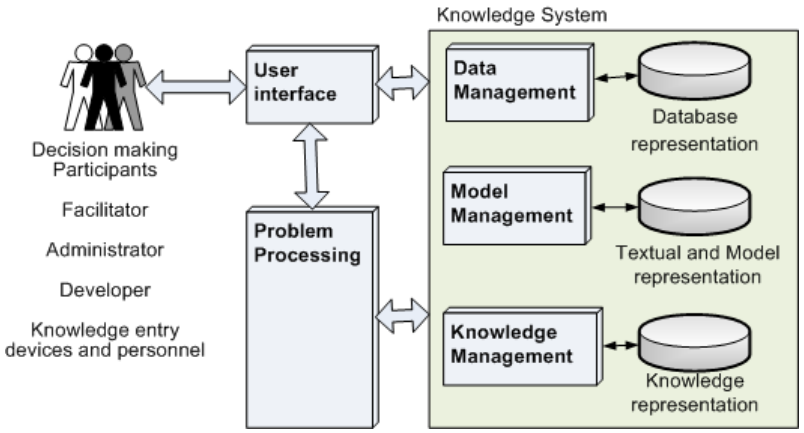

Figure 1- Top-level architecture of VirtualECare GDSS

The VirtualECare GDSS (Figure 1) has a rather 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's activity can be triggered by events that are detected outside de system or inside the system. The Problem Processing module incorporates the ability to evaluate the quality of knowledge available for a decision process.

3 Knowledge Representation and Reasoning

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. The generality of logic programs represents implicitly negative information, assuming the Closed-World Assumption (CWA) [6].

An extended logic program (ELP), on the other hand, is a finite collection of rules of the form [7]:

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, (Proceedings of the International Symposium on Distributed Computing and Artificial Intelligence (DCAI 2009), Salamanca, Spain, 2009), 2009.

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

$$?\, p_1 \wedge ... \wedge p_m \wedge not\ p_{m+1} \wedge ... \wedge not\ p_{m+n} \quad (2)$$

where **?** is a domain atom denoting falsity, the $\boldsymbol{p_i}$, $\boldsymbol{q_j}$, and $\boldsymbol{p}$ are classical ground literals, i.e. either positive atoms or atoms preceded by the classical negation sign ¬.

We need to represent explicitly negative information, as well as directly describe the CWA for some predicates. Three types of answers to a given question are then possible: *true*, *false* and *unknown*. 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(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 1 - Representation of knowledge about the user environment

The symbol ¬ represents the strong negation, denoting what should be interpreted as false, and the term *not* designates negation-by-failure. The second clause represents the environment temperature of another room, the *living_room*, has not, yet, been established. The symbol ⊥ represents a null value of an undefined type. It is a representation that assumes any value as a viable solution. It is not possible to compute, from the positive information, the value of the environment temperature of the *living_room*. The fourth clause of Program 1 (the closure of predicate env_temp) discards the possibility of being assumed as false any question on the specific value of environment temperature for *living_room*.

The value of the environment temperature for *dining_room* is foreseen to be 20, with a margin of mistake 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. As a different case, let's consider the environment

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, (Proceedings of the International Symposium on Distributed Computing and Artificial Intelligence (DCAI 2009), Salamanca, Spain, 2009), 2009.

temperature of the *dining_room*, that is unknown, but one knows that it is specifically 22 or 25.

The Quality of Information (QK) with respect to a generic predicate P is given by $QK_P = 1/Card$, 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 the inverse of the sum of the possible combinations of exceptions.

$$QK_P = \frac{1}{C_1^{Card} + \cdots + C_{Card}^{Card}} \qquad V_i(x) = \sum_{j=1}^{n} w_{ij} * V_{ij}(x_j) \qquad (3)$$

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: w_{ij} stands for the relevance of attribute j for predicate i (it is also assumed that the weights of all predicates are normalized. It is now possible to define $\boldsymbol{V_i}$ as a scoring function for a value $x = (x_1, ..., n)$ in the multi dimensional space defined by the attributes domains.

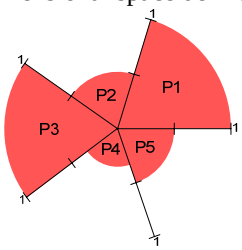

Figure 2 - A measure of the quality of knowledge for a logic program or theory P

It is now possible to measure the QK that occurs as a result of a logic program, by posting the $V_i(x)$ values into a multi-dimensional space and projecting it onto a two dimensional one, as we see in Figure 2, for five predicates [8].

4 Group Decision Support Systems

Group Decision Support Systems (GDSS), also called Multiparticipant Decision Support Systems (MDSS), have been the subject of much research, have matured over a period of many years and there are many examples of their successful application [9-11]. The main characteristic of many GDSS implementations is a Problem-Processing System (PPS) [9] with the ability to provide strong coordination for handling or even guiding participant interactions, linked with abilities of knowledge acquisition from participants, incorporating this knowledge into the Knowledge System (KS), which serves as group memory.

The VirtualECare GDSS is a knowledge-driven or intelligent DSS [12] based on an inference engine with rules, although it also relies on database and model representations. The use of an inference engine with rules is the most common development environment for knowledge-driven decision support systems [11]. Rules

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, (Proceedings of the International Symposium on Distributed Computing and Artificial Intelligence (DCAI 2009), Salamanca, Spain, 2009), 2009.

are easy for managers and domain experts to understand and it is easier to provide explanations to users of the DSS. Also, it can combine information about uncertainty in conclusions with rules.

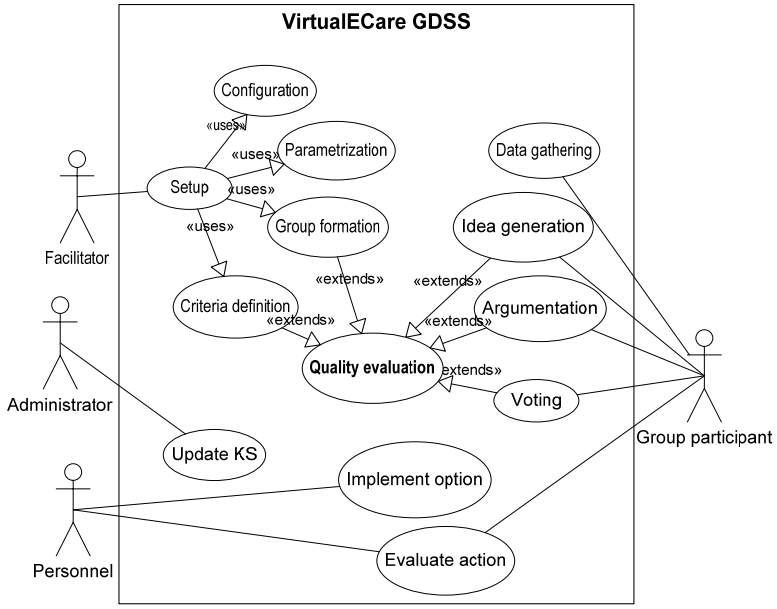

Figure 3 – Use Case view of VirtualECare GDSS

4.1 Context of Decision Making in VirtualECare

We make the distinction between *non-cooperative multi-member decision making* and *cooperative group decision making* [12]. In VirtualECare the context of decision making is the *cooperative group decision* one. Another characteristic of decision making in VirtualECare is that there is not a hierarchic team structure decision. No individual participant has the authority to make a specific decision. In contrast, all the participants share the same interest in the decision outcome and have an equal say in the decision formation.

The decision model of the VirtualECare GDSS is the *rational model*, based on objectives, alternatives, consequences and search of optimal. This model assumes that the decision maker knows all (or most of) the alternatives, their associated information and the consequences of every choice, at least the short term ones. The alternative that provides the maximum utility, i.e. the optimal choice, is then selected. It is also assumed that the participants assess the pros and cons of any alternatives with specific goals and objectives in mind. It is not new that, besides improving group communication activities, a GDSS must provide a group centered problem solving environment, aimed for helping decision makers consider uncertainty, form preferences, make judgments and take decisions [13]. Figure 3 depicts a Use Case view of the VirtualECare GDSS, showing a central use case “Quality evaluation”.

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, (Proceedings of the International Symposium on Distributed Computing and Artificial Intelligence (DCAI 2009), Salamanca, Spain, 2009), 2009.

4.2 Problem Solving

The staged nature of decision making processes is established by several studies [14, 15]. The models of real-world decision procedures includes time-divided and / or single time decision periods, where content homogeneous and content heterogeneous operations are performed [15]. The VirtualECare GDSS follows this procedural staged nature, as we can see in Figure 4.

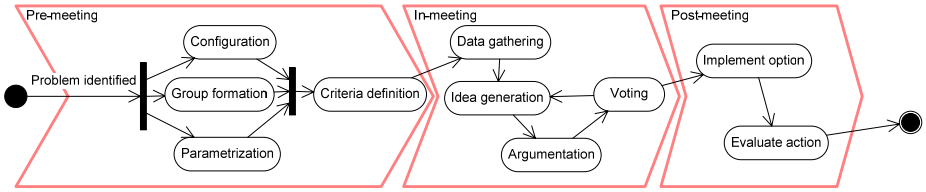

Figure 4 – Staged decision process model

Traditionally, Rational Choice Theory (RCT) is applied to decision support systems which follows the prescription of Herbert Simon [16], where the agent only "satisfices" its expected utility, rather than optimizing or maximizing it. Either way, Simon prescribes a linear decision making process, moving through three stages: *intelligence*, *design* and *choice*. *Intelligence* involves the perception and diagnostic of the problem, searching for the conditions that call for decisions. *Design* concentrates upon inventing, developing and analyzing possible courses of action, defining goals and criteria. Finally, the *Choice* stage concentrates upon selecting an alternative identified in the previous phase.

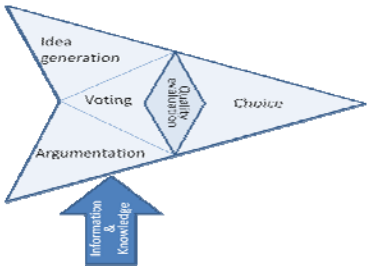

Figure 5 – In-meeting stage: design and choice phases separated by quality evaluation

The underlying process model of the VirtualECare GDSS follows Simon's empirical rationality. The *intelligence* stage occurs continuously, as the GDSS interacts with other components of 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 participants.

The *design* and *choice* phases occur in the in-meeting stage (see Figure 4). The in-meeting stage cycles through several iterations, similarly to the *circular logic of choice* of Nappeelbaum [17]. 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 with Jones and Humphreys model of the Decision Hedgehog [14]. Instead of constructing

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, (Proceedings of the International Symposium on Distributed Computing and Artificial Intelligence (DCAI 2009), Salamanca, Spain, 2009), 2009.

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.

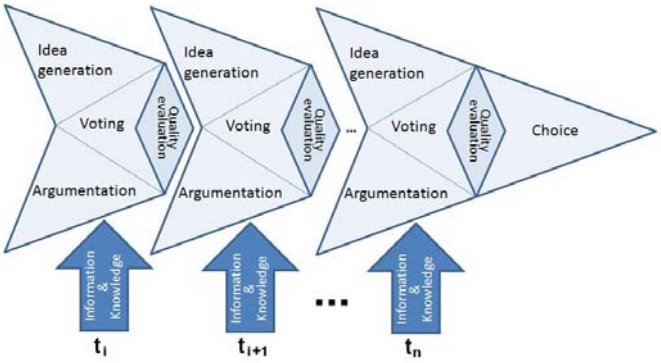

Figure 6 – In-meeting stage with several iterations

This way, the quality of information is evaluated within each iteration, for every possible pathway. The knowledge system is scanned for the needed information with a previously agreed threshold of measured quality [8]. If the quality of information does not reach the necessary threshold, new information and/or knowledge is acquired to the knowledge system and the process restarts. Figure 5 illustrates this process for a single iteration and Figure 6 depicts the situation when the quality threshold is not reached until the nth iteration, when the decision is made.

Even when time compels the group to make a decision before the quality threshold is reached, the quality evaluation is useful to assess and record the context in which the decision was made.

5 Conclusions

As a result of this work, we present a process model for group decision making where the quality evaluation of information plays a central role. We use an Extended Logic Programming language for the representation and reasoning with imperfect information. We also present an architecture of a Group Decision Support System in the context of VirtualECare project, a system aimed at sustaining online healthcare services.

The decision process model is a staged one, with several interactions, with the progress being controlled by the quality evaluation of the available information. If the quality of information does not reach a previously defined threshold, the system advises to collect more accurate information before progressing.

In a future development, the system will be able to make recommendations on how to progress in the decision making, using a Case Based Reasoning (CBR) approach. The case memory will represent past decision making situations, where we can find the most adequate types of information and origin.

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, (Proceedings of the International Symposium on Distributed Computing and Artificial Intelligence (DCAI 2009), Salamanca, Spain, 2009), 2009.

References

1. Costa, R., et al., *VirtualECare: Group Decision Supported by Idea Generation and Argumentation*, in *The 9th IFIP Working Conference on Virtual Enterprises (PRO-VE 2008)*. 2008: Poznan, Poland.
2. 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.
3. Apt, K.R. and R. Bol, *Logic Programming and Negation: A Survey.* Journal of Logic Programming, 1994. **19**: p. 9-71.
4. 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 Verlag. p. 402-436.
5. Zadeh, L.A., *Fuzzy logic.* Scholarpedia, 2008. **3**: p. 1766.
6. Hustadt, U. *Do we need the closed-world assumption in knowledge representation?* in *KI'94 Workshop*. 1994. Saarbrüken, Germany: Baader, Buchheit, Nutt (eds.).
7. 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.
8. Lima, L., et al., *Quality of Information in the context of Ambient Assisted Living*, in *Advances in Soft Computing*. 2008, Springer- Verlag. p. 624-633.
9. Burstein, F. and C.W. Holsapple, eds. *Handbook on Decision Support Systems: Basic Themes*. International Handbooks on Information Systems. 2008, Springer.
10. Sprague, R.H. and E.D. Carlson, *Building Effective Decision Support Systems*. 1982: Prentice-Hall.
11. Power, D.J., *Decision Support Systems: Concepts and Resources for Managers*. 2002: Greenwood Publishing Group.
12. Lu, J., et al., *Multi-objective Group Decision Making: Methods, Software and Applications with Fuzzy Set Techniques.* Series in Electric and Computer Engineering. 2007: Imperial College Press.
13. Humphreys, P. and E.L. Nappelbaum, *Structure and communications in the process of organisational change: Eastern European experience and its general relevance*, in *Decision Support in Organizational Transformation: Proceedings of the IFIP TC8 WG8.3 International Conference on Organizational Transformation and Decision Support*, P. Humphreys, et al., Editors. 1997, Springer: La Gomera, Canary Islands.
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*. 2005. Paris (France): CEUR-WS.
15. Kolbin, V.V., *Decision Making and Programming*. 2003: World Scientific.
16. Simon, H.A., *Models of Bounded Rationality: Empirically Grounded Economic Reason*. Vol. 3. 1982: MIT Press.
17. Nappelbaum, E., *Systems logic for problem formulation and choice*, in *Decision Support in Organizational Transformation: Proceedings of the IFIP TC8 WG8.3 International Conference on Organizational Transformation and Decision Support* P. Humphreys, et al., Editors. 1997, Springer.