A MULTI-CRITERIA DECISION AID AGENT APPLIED TO THE SELECTION OF THE BEST RECIPIENT IN A TRANSPLANT

Aïda Valls, Antonio Moreno, David Sánchez
Working Group on Multi-Agent Systems (GruSMA), Research Group on Artificial Intelligence
Computer Science and Mathematics Department, Universitat Rovira i Virgili
Address: Campus Sescelades, Avda. Països Catalans, 26, 43007 Tarragona, Spain
Email: avalls@etse.urv.es

Keywords: Multi-Agent Systems, Decision Support Systems, Multi-Criteria Decision Aid, Medicine, Transplants.

Abstract: In this paper we describe an agent that applies a new multi-criteria decision methodology to analyse and rank a list of possible recipients for a particular organ. The ranking obtained is of great help for the Hospital Transplant Co-ordinator who has to make the final decision of which patient receives the organ. The agent that we have designed and implemented can be used in any other similar problem in which we have a list of alternatives that are evaluated with several qualitative preference criteria.

1. INTRODUCTION

Making good decisions is an important issue in all domains. People who have this responsibility are very keen on the help of decision support systems. When we have a set of possible alternatives, each of them described by a set of criteria, we are faced with a “Multiple Criteria Decision Problem”. MCDA (multi-criteria decision aid) methods are being developed for this particular case since the early 70s; some well-known methodologies are ELECTRE, PROMETHEE, STEM and AHP (see an introduction to them in (Bana e Costa, 1990)). At the beginning they were designed to work with numerical values, but nowadays there are some attempts to introduce qualitative valuations (Delgado et al., 1992). The aim is to allow the experts to express their knowledge (i.e. opinions, preferences) in a comfortable and easy to understand way rather than forcing them to change the valuation scales they are used to.

In multi-agent systems, MAS, (Weiss, 1999) we usually delegate to our agents capabilities that imply a multi-criteria decision making process. Imagine that our personal agent has searched for opinions on 50 European ski resorts in order to recommend us only 3 of them. He must apply a MCDA technique to be able to select those that fit best with our preferences (in terms of economic cost, distance to be travelled, difficulty level of the tracks, etc.). As these techniques are not simple, we propose to delegate the analysis and ranking of the alternatives to an MCDA analyst agent, while our personal agent can continue doing other activities for us.

In this paper we argue that it may be useful to generate agents that are experts in solving MCDA problems. These agents would receive requests of any other agent that has to face a problem like this, regardless of the particular application or multi-agent system to which it belongs.

In this paper we present a working prototype of a MCDA analyst agent that applies a new methodology for qualitative and numerical values. This agent has been used to help in the selection of the best recipient for an organ transplant. In section 2 we introduce this multi-criteria problem and we show the multi-agent system designed to help in all the transplant co-ordination steps. In section 3 we describe in detail the MCDA methodology followed by the agent. The feasibility and interest of agentifying this kind of methods is discussed in section 4. In section 5 we comment some aspects of the prototype. Finally, some conclusions and future research work are explained.
2. A MULTI-AGENT SYSTEM FOR TRANSPLANT CO-ORDINATION

2.1 The Spanish transplant co-ordination model

The process that has to be followed in order to co-ordinate all the people and information involved in a transplant is quite complex. In the case of organ transplants, this process must be done in a very short period of time (only some hours) because they cannot be frozen as we do with tissues and bones. In collaboration with some hospitals we are designing a Multi-Agent System to assist in all the steps previous to the organ transplant operation. The system will follow the national and international rules established for transplants and will try to model the process that is done at the moment, known as the Spanish co-ordination model (Matesanz et al., 1995). This model is centred in the figure of the Hospital Transplant Co-ordinator (HTC) of each hospital. In each hospital a list of people waiting for an organ is maintained. When a donor is recognised, the transplant process starts. The HTC of the hospital is informed and he starts the search of a suitable receiver for the organ, following the predefined hierarchy shown in fig. 1. This searching process allows to find the receiver that is closer to the hospital that has the organ, in order to minimise the transport time and the associated costs. At the regional and zonal levels there exists a predefined consulting order that tries to ensure a fair distribution of the organs.

In figure 1 we have represented a portion of the Spanish hierarchy. We can see that Spain is divided into 6 zones that group the 19 Spanish autonomous regions according to their location.

In our MAS we will respect this hierarchical organisation in order to find a receiver. The use of computers and Internet will provide: (i) high speed in the communications with other hospitals and (ii) the possibility of analysing a larger amount of data. These two aspects imply that we will have more time to carry the organ from one city to another, which means that the distance will not be as important as it is now.

2.2 An intra-hospital multi-agent system

We have also proposed to use a Multi-Agent System to deal with the transplant activities that have to be held within the donor’s hospital. In figure 2 we can see the architecture of the internal MAS of a hospital. The dotted square contains the agents that belong to the same hospital.

The agent that is in contact with the medical personnel (and, in particular, with the hospital transplant co-ordinator) is called TCA (Transplant Co-ordinator Agent). This agent receives the characteristics of the donor and the organ that can be transplanted and starts the process to recommend the best possible receivers. First of all, he searches in the local database for potential receivers at the same hospital. This search is made with the help of the Medical Database Wrapper, which is the agent that is in charge of the access to the hospital database. If this local search is unsuccessful, TCA sends a request for other candidates to the other TCAs in the same region (via the regional co-ordinator). If no adequate receiver is found, he continues the search following the hierarchical organisation shown in fig.1 (at the zonal level and, if necessary, at national level).

When TCA has obtained a list of candidates, he sends them to the Transplant Specialist (TS) agent, which has knowledge about the field of organ transplants. This agent discards the patients that do not fulfil some basic compatibility conditions (e.g. the blood types of the donor and the potential receiver are not compatible).
After this initial filtering, TS compares the attributes of the donor with the ones of the candidates and builds a preference matrix using qualitative criteria. In the initial prototype TS only takes into consideration 6 attributes of the receivers, which can be expressed with numerical or qualitative values:

- weight
- size of the organ needed
- antigens
- age
- location of the hospital where the transplant should be performed
- amount of time that the person has been waiting for the organ

TS analyses each attribute separately and builds a preference criterion according to the similarity with the donor. The preferences are expressed with linguistic values in predetermined vocabularies that are easy to interpret for the Hospital Transplant Coordinator (see table 1). The final criteria are:

- difference between the weights of the donor and the receiver
- difference between the size of the donor’s organ and the size of the organ needed by the receiver
- number of different antigens
- difference of age between the donor and the receiver
- distance to cover to bring the organ to the receiver
- amount of time that the receiver has been waiting for this organ

<table>
<thead>
<tr>
<th>Attribute</th>
<th>worst value</th>
<th>. . .</th>
<th>best value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>inadequate, feasible, good, optimum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Size</td>
<td>inadequate, feasible, good, optimum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Antigens</td>
<td>different, similar, identical</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Age</td>
<td>more_thn_20, more_thn_17, more_thn_14, more_thn_11, more_thn_8, more_thn_5, the_same</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distance</td>
<td>country, zone, region, city, hospital</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Waiting time</td>
<td>very_short, short, acceptable, long, very_long</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Vocabularies of the qualitative criteria.
There exist many different MCDA techniques; however, we are interested in distinguishing those that work with numerical values and those than work with qualitative ones (i.e. linguistic terms). The qualitative approach is required when conventional numerical scales cannot be applied to a criterion (e.g. mental disposition to accept a transplant). In other cases, it is only used to deal with uncertainty or lack of precision in the data (e.g. great accuracy to express the distance between the donor and the receiver is not needed). In fact, we consider that the 6 criteria mentioned above must be compared without taking into account the precise number, but the semantics behind it (e.g. we want to distinguish if the receiver is in the same hospital than the donor, in another hospital of the same city, in another city of the same region, in another region of the same zone or in other zones of the country, but we do not need to consider the exact distance in kilometres between them).

The preferences constructed with the vocabularies shown in Table 1 are sent to an MCDA agent that ranks the receivers from the best to the worst (this ordering process is explained in detail in the following section). The result is then sent back to TCA, that displays it along with some qualitative values that may help the decision maker to finally select the most appropriate receiver of the organ.

3. THE MULTI-CRITERIA DECISION AID METHODOLOGY

After observing that MCDA methods can usually deal only with a unique type of values (i.e. numerical, fuzzy, ordinal), we have implemented a methodology that accepts that each criterion uses the most appropriate type of values according to its meaning. That is, the decision matrix that our method processes may have mixed types of values. At the moment, we have concentrated on numerical criteria using different scales and qualitative criteria using different vocabularies (i.e. different sets of possible terms). Our decision methodology consists of two phases: (i) applying a clustering procedure to aggregate the criteria (the result obtained is a set of groups of alternatives, i.e. a set of classes) and (ii) ordering these classes using a distance function to the ideal alternative. These steps are described in the following subsections.

3.1 Aggregation of criteria

The classification process has two phases:
(a) The building of a similarity matrix, that contains the measures of proximity between each pair of alternatives. Several similarity or dissimilarity functions can be used, as differences (e.g. Manhattan, Hamming, Mahalanobis), association coefficients (e.g. Jaccard, Dice) or correlation measures (e.g. Pearson). Each measure has different properties, and it is not possible to determine which is the best for a particular set of data.
(b) The building of a set of classes that groups those objects that are more similar. There are different methods to be applied, but, up to now, it is impossible to define a way to choose neither the best method, nor the best for a particular problem. We can distinguish two families of methods, with many different algorithms developed under each approximation.

Hierarchical clustering methods: classes are embedded forming a tree. The root is the most general class which contains all the objects (i.e. alternatives), and the leaves are the most specific groups, that contain a unique alternative.

Partitional clustering methods: classes are mutually exclusive. The clusters are formed optimising a ‘clustering criterion’.

Due to the impossibility of determining the best methodology (similarity measure + grouping technique), we have built a classifier that incorporates different clustering methods. However, we have concentrated only in hierarchical non-overlapping methods. The system admits missing values, that is, if the information supplier can not give his opinion for an alternative, he can leave this alternative unvalued.

The clustering process starts with the building of a similarity matrix that contains the result of comparing each pair of alternatives. This comparison is done using different weights for each criterion, to indicate their relative importance. This similarity is obtained by applying functions that
have been adapted to work with different kinds of values and produce a numerical result. Considering that the lower the value the more similar the alternatives are, we search for all the pairs of alternatives that have the lowest value in the similarity matrix; each of these pairs of alternatives forms one initial class. Then, all the alternatives that belong to one of these classes are removed from the similarity matrix. Finally, these new classes are incorporated into the similarity matrix. The similarity of the new classes to the other classes/alternatives can be measured using different methodologies, such as single linkage, complete linkage or centroid clustering (Jain et al., 1988).

Then, we repeat the process until we have obtained a number of classes similar to the granularity of the vocabularies of the initial criteria, or we have spent all the time that we were allowed to use. The result obtained is a nested partition (the clustering process may be repeated until we have a single tree).

For example, let us imagine a transplant situation with 7 possible receivers (A-G). We will take into consideration three qualitative criteria that express our preference against different concepts: Weight, Size and Waiting Time (the vocabularies of these criteria were shown in table 1). In table 2 we have the decision matrix.

<table>
<thead>
<tr>
<th>Weight</th>
<th>Size</th>
<th>Waiting Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>inadequate</td>
<td>short</td>
</tr>
<tr>
<td>B</td>
<td>good</td>
<td>feasible</td>
</tr>
<tr>
<td>C</td>
<td>good</td>
<td>very_long</td>
</tr>
<tr>
<td>D</td>
<td>feasible</td>
<td>acceptable</td>
</tr>
<tr>
<td>E</td>
<td>good</td>
<td>long</td>
</tr>
<tr>
<td>F</td>
<td>optimum</td>
<td>acceptable</td>
</tr>
<tr>
<td>G</td>
<td>good</td>
<td>optimum</td>
</tr>
</tbody>
</table>

Table 2: Decision matrix with 3 criteria.

If we compute the similarity using the values in table 2, we obtain the matrix in table 3, which is a triangular matrix. In this example, the minimum value that we can find in this similarity matrix is 0.2, which corresponds to the pair [C, E]. Thus, this will be the first class generated (Class1). We include it into the similarity matrix after eliminating the individual alternatives C and E (see table 4).

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>1.3</td>
<td>0.4</td>
<td>1.2</td>
<td>1.3</td>
<td>1.3</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>0.3</td>
<td>0.7</td>
<td>0.5</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>C</td>
<td>-</td>
<td>-</td>
<td>0.4</td>
<td>0.2</td>
<td>0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>D</td>
<td>-</td>
<td>-</td>
<td>0.8</td>
<td>0.9</td>
<td>0.9</td>
<td>-</td>
</tr>
<tr>
<td>E</td>
<td>-</td>
<td>-</td>
<td>0.5</td>
<td>0.5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>F</td>
<td>-</td>
<td>-</td>
<td>1.1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>G</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3: Similarity matrix of data in table 2.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>D</th>
<th>F</th>
<th>G</th>
<th>Class1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-</td>
<td>1.1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>0.7</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>-</td>
<td>0.9</td>
<td>0.9</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>-</td>
<td>1.1</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Class1</td>
</tr>
</tbody>
</table>

Table 4: Similarity matrix after merging C and E into Class1.

At the second step, the lowest value is 0.4, which tells us to join A and D in one class, and B and Class1 into another class. The result is the partition: [{F}, {G}, {A,D}, {B,C,E}]. In this case, the clustering process finishes because we have 4 classes and this is, approximately, the number of terms in the vocabularies of the initial criteria. The similarity value that we use to decide whether to form a class or not is an indicator of the intra-cluster cohesion. The higher the value, the more spread the alternatives in the class are. With this information we can see that {A,D} has a cohesion level of 0.4, the same as {B,C,E}. This can be used as a measure of the goodness of the result. After the ordering, if the value of the cohesion of the best class is high it means that inside the class there are alternatives that are better than the others; however, if the cohesion value is low, it means that all the alternatives are almost identical. In the first case, the decision maker may perform an analysis of the alternatives within the class in order to distinguish among them, whereas in the second case it would not be worth making that analysis, because the differences are not significant enough.
decision maker wants to perform a more detailed study of the first class, we can give him the sequence of nested classes that have been formed before stopping the clustering. This tree can be cut at another level and the classes obtained can then be ordered again to distinguish the best ones.

### 3.2 Ranking the alternatives

The ordering process is based on the similarity of the prototypes of the classes to an *Ideal* alternative (usually a fictitious one). To compute this similarity we use the same function applied in the clustering. In the example, the *Ideal* alternative will have the best term in each attribute, that is Weight=optimum, Size=optimum and Waiting Time=very_long. In table 5 we can see the distance between the *Ideal* and the prototypes of the classes obtained after the clustering (table 4), and the projection of these values into the unit interval. We also show in table 5 the distance of the *Ideal* alternative to the worst possible alternative (with the worst term in each attribute).

<table>
<thead>
<tr>
<th>F</th>
<th>G</th>
<th>A+D</th>
<th>B+C+E</th>
<th>Worst</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9</td>
<td>0.5</td>
<td>1.7</td>
<td>0.73</td>
<td>2.4</td>
</tr>
<tr>
<td></td>
<td>Translation into [0,1]</td>
<td>0.375</td>
<td>0.208</td>
<td>0.708</td>
</tr>
</tbody>
</table>

Table 5: Similarity to the *Ideal* alternative.

From the values in table 5, we have this partial ordering from the best alternative to the worst one: G, B=C=E, F, A=D. This result is not given directly to the user. The last step consists of choosing a qualitative term to describe each of the classes from a pre-defined vocabulary: {ideal, very_good, good, recommendable, acceptable, not-recommendable, bad, terrible}. The human experts in the field considered that 8 terms should be enough to describe the suitability of the alternatives.

To assign a term to each class, we look at the similarity value among each class and the *Ideal* alternative. We have divided the unit interval into 8 parts and each one corresponds to one term as shown in table 6.

<table>
<thead>
<tr>
<th>Ideal</th>
<th>[0.0,0.125]</th>
</tr>
</thead>
<tbody>
<tr>
<td>very_good</td>
<td>(0.125,0.25]</td>
</tr>
<tr>
<td>good</td>
<td>(0.25,0.375]</td>
</tr>
<tr>
<td>recommendable</td>
<td>(0.375,0.5]</td>
</tr>
<tr>
<td>acceptable</td>
<td>(0.5,0.625]</td>
</tr>
<tr>
<td>not-recommendable</td>
<td>(0.625,0.75]</td>
</tr>
<tr>
<td>bad</td>
<td>(0.75,0.875]</td>
</tr>
<tr>
<td>terrible</td>
<td>(0.875,1]</td>
</tr>
</tbody>
</table>

Table 6: Correspondence between similarity and final vocabulary.

If we assign directly a term to each class according to the similarity value in table 5, we have that class {B,C,E} and class {F} are both “good”. In order to avoid that more than one class receives the same term, which would make them indistinguishable, we apply the following algorithm:

**Algorithm explain_result**

1. **k := number of classes to be explained**
2. **Take the best class of the ranking (C_best)**
3. **While k>0 do**
   1. **Take all those terms in the vocabulary that have at least k-1 worse terms \([t_a..t_b]\). Moreover, \(t_a\) should not be better than any previously assigned label.**
   2. **If similarity(C_best, Ideal) belongs to one of the intervals of the terms in \([t_a..t_b]\) then**
      1. **C_best takes the term corresponding to this interval**
      2. **else**
         1. **if C_best should use a better label then**
            1. **C_best takes \(t_a\) (the best possible label)**
         2. **else**
            1. **C_best takes \(t_b\) (the worst possible label)**
      3. **end if**
   3. **end if**
   4. **k := k-1**;
   5. **If \(k = number of terms that follow the assigned term then**
      1. **Assign these k labels to the k remaining classes**
      2. **else**
         1. **Take the class that follows C_best in the ranking, and call it C_best**
      3. **end if**
   6. **end while**
4. **end algorithm.**
Let us apply this algorithm to our example. There are 4 classes (k=4), so the initial set of possible terms is \{ideal, very_good, good, recommendable, acceptable\}. The best class is \(C_{best} = \{G\}\), with similarity 0.208, which belongs to the interval \((0.125,0.25]\) that corresponds to the term \emph{very_good}. After this assignment, \(C_{best} = \{B,C,E\}\), with similarity 0.304. Now, the set of possible terms is \{good, recommendable, acceptable, not-recommendable\}. We choose the term \emph{good} with interval \((0.25,0.375]\). At the following step, \(C_{best} = \{F\}\), with similarity 0.375, and the possible terms are \{recommendable, acceptable, not-recommendable, bad\}. As 0.375 does not belong to any of the intervals of these terms, because it is greater, we choose the best possible term, \emph{recommendable}, for class \{F\}. Finally, \(C_{best} = \{A,D\}\), with similarity 0.708, and the available terms are \{acceptable, not-recommendable, bad, terrible\}. In this case 0.708 belongs to the interval corresponding to \emph{not-recommendable}. Table 7 shows the final result.

<table>
<thead>
<tr>
<th>very_good</th>
<th>good</th>
<th>recommend.</th>
<th>not-recommendable</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>B,E</td>
<td>F</td>
<td>A,D</td>
</tr>
</tbody>
</table>

Table 7: Ranking of the MCDA method.

This MCDA methodology is the one that follows our agent, called \emph{ClusDM} (clustering for decision making). The characteristics of this agent are explained in the following section.

4. ClusDM: A MCDA AGENT

The \emph{agentification} of MCDA methods allows us to release other agents from the decision process. If an agent needs to make a decision analysis, he will delegate this task to the MCDA agent, while he can be busy performing other tasks. This agent specialisation is useful because multi-criteria decision aid is not a simple task. In the real world, it is usually done by experienced analysts who know how to apply the methodology and how to interpret the obtained results. Moreover, not all the MCDA methods can be successfully applied to the same kind of problems, it depends on the properties of the method and the characteristics of the problem. Thus, the first question that the analyst has to solve is the selection of which MCDA technique to apply. In a multi-agent system we could have different MCDA agents, where each of them was an expert in a particular technique.

An agent (Wooldridge et al., 1995) is a computer system capable of flexible autonomous action in some environment. The main properties of agents are:

- \emph{Social ability}: an agent must be able to communicate with other agents, and cooperate with them to solve complex tasks.
- \emph{Reactivity}: an agent is aware of the changes in the environment and responds to them in a timely fashion.
- \emph{Autonomy}: the agent may decide whether to fulfil a given request or not, and may also decide which is the best way to try to achieve his goals.

\emph{ClusDM} is an agent that offers a very specific service: ranking a set of alternatives using multiple criteria. So, this agent may receive a message with the request of performing a decision analysis, and he will return a message that informs the requester of the obtained result. As shown in previous sections, this agent must be able to communicate with the other agents engaged in the complex process of organ transplant management. Concerning the second property, reactivity, ClusDM aborts the clustering process if the time at his disposal is near to expire, in which case the result is the ordering of the classes built up to that moment. On the other hand, if it detects that the result will not be good (because the intra-cluster cohesion has high values, or the difference between the number of elements in each class is significantly high), ClusDM aborts the process and communicates it to the requester. At the reception of a request, ClusDM can decide if he will make the ranking or not, depending on the characteristics of the message received (the data matrix is correct, the information about the semantics of the criteria is correct and the amount of data is tractable); thus, it also shows a certain degree of autonomy.
5. THE PROTOTYPE

The Multi-Agent System shown in fig. 2 has been implemented using Jade (Bellifemine et al., 1999). Jade is a collection of Java libraries that ease the implementation of FIPA-compliant\(^1\) multi-agent systems. We are running this prototype on Windows in standard PCs, although it could be used in any other platform that supports Java. In fig. 3 we can see the interface that the Transplant Co-ordinator Agent shows to the hospital transplant expert. Before requesting the ranking of the set of patients, the user must assign a weight to each criterion. After processing the patients data (which is stored in the file indicated at the top of the window) the system will display the result at the bottom. Then, the user can express his agreement with this result using the vocabulary listed at the bottom-right side.

![Figure 3: Transplant Co-ordinator Agent Interface.](image)

In this example, the system is displaying the result given by the ClusDM agent, which has used the Manhattan distance to obtain the similarities.

6. DISCUSSION AND FUTURE WORK

In the problem of organ transplant co-ordination the use of decision support techniques can help the person in charge of the process to make a better selection more quickly. The automatisation of filtering and ranking of the patients along with the searching of possible receivers can save a precious time in this critical problem. However, if the result is not clear or the user does not rely on it, the system will be misused. For this reason we have proposed a methodology that gives a linguistic label to each patient, which expresses his goodness for the transplant. A quality value is also assigned to each ranking (based on the intra-cluster cohesion), which can tell to the Hospital Transplant Co-ordinator if the system has been able to sort out the problem successfully.

In complex problems where different tasks should be performed, and distributed sources of information should be queried, the use of Multi-Agent Systems is appropriate. Sometimes, agents will need to solve a multi-criteria decision problem. We propose to have agents dedicated to this task. In this paper, we have explained a new multi-criteria decision methodology that can be agentified. The characteristics of this agent have been reviewed in section 4. However, other MCDA methodologies could also be agentified. Then, the problem of choosing the MCDA agent to use in each MAS has to be studied.

REFERENCES


---

\(^1\) FIPA (Foundation for Intelligent Physical Agents) is a non-profit association that provides internationally agreed specifications for developing agent-based applications (FIPA, 2001).