Cooperative case-based agents for acute stroke diagnosis

Beatriz Lopez [1], **Carles Pous** [2], **Joaquin Serena** [3] and **Jordi Piula** [4]

Abstract.
This paper presents a multi-agent case-based system designed to support acute stroke diagnoses and management. Even though an acute stroke can be diagnosed by physicians, the final clinical category and treatment of the stroke often depends on the skills and experience of neurologists in using new technology. From the Spanish Neurological Society a data base of experiences in hospitals have been collected in the last years. From this database, we have designed case-based agents to support diagnosis. These case-based agents keep information about experiences in a single hospital and outline the particular decision criteria employed by the main physician. They collaborate when they show a lack of confidence in the isolated decision problem. In this paper both the multi-agent collaboration mechanism and the case-based decision making process are described.

1 INTRODUCTION

Acute strokes are medical emergencies that require diagnoses from expert neurologists in order to detect the illness in the appropriate therapeutic time window [5]. Thanks to the development of new treatments such as the rtPA treatment, mortality rates have decreased in the last decades. However, the final diagnosis of the patients in often imprecise. That is, patients can be diagnosed as having an acute stroke, but the clinical category is often *undetermined*.

This indetermination is due to two main reasons. First, the clinical category is established according to the definition recorded at either the Lausanne [4] or the TOAST [6] classification, which were established in the 80s based on the technology of that time. For instance, the TOAST classification consists of five categories: atherothrombothic, cardioembolic, small vessel disease, other (infrequent causes), and undefined (stroke of unknown cause), based mainly on the results of stroke symptoms, cranial-TC scan, ultrasounds studies of supraaortic trunks, and echocardiography (ECG). Current technology can be more precise in characterizing each patient's situation. A significant amount of the *undefined* cases can be re-considered in other clinical categories after obtaining the results from a RM or ECO image. In addition, recent technology helps the neurologist to understand the causes of the illness instead of the stroke mechanism (which is provided by the classification), which allows the patient's treatment to be individualized. On the other hand, Stroke Data Banks, and particularly BADISEN, include a considerable number of variables not considered to classify stroke using traditional criteria but essential in the correct management of patients. Furthermore, Stroke Data Bank are continuously updated, adding new variables of interest in stroke (e.g. data from new technologies such as MRI, new therapies, causes of stroke discovered recently, etc.). Thus, the final diagnoses and treatment of the patient depends on the skills and experience of the neurologist in using recent technologies that are not yet gathered in any medical classification. Therefore, different criteria can arise between two physicians of different health centres according to the interpretation, experience and knowledge of previous clinical studies in which recent technology are applied.

The second cause of the indetermination is related to the first, since less-experienced neurologists could bypass some patient data that facilitates the clinical category due to their lack of experience in using the new technology. Even more, in any classification of stroke, the same category include different causes of stroke (e.g. the category "cardioembolic" include any cause of stroke of cardiac origin such us acute myocardial infarction, atrial fibrillation, congenital or hereditary cardiac diseases etc.).

Given this situation, we have considered the possibility of developing a computerized support system for supporting the acute stroke diagnosis. The idea is to use past experiences to guide current diagnoses. Therefore, if a diagnosis is unknown, but the patient data is similar enough to a past experience, the differences between the current and past patient data can be used to benefit the current diagnosis and particularly in indicating the best treatment for the patient, an aspect that is continuously changing and is not considered in the classification of stroke. If the difference consists in unknown data about the current patient this can be easily checked with either an additional patient exploration or by revising the patient's history.

An experience-based approach is feasible thanks to the stroke database the Spanish Neurological Society has been gathering over the last few years. Around one thousand acute stroke cases have been stored in the BADISEN Stroke Data Bank [3]. In Figure 1 there is a snapshot of the tool used by physicians to gather the data. Analysing this information showed that when an acute stroke is diagnosed, approximately 30% of them have an unknown clinical category using the TOAST criteria, the most widely and accepted protocol used to classify stroke.

However, diagnosis based on past experience solves one of the first problems concerning the lack of clinical category information. We believe the second problem concerning the different medical criteria can also be tackled by using a multi-agent system. Each agent keeps the experiences of a physician or hospital that follows a given set of criteria, and all agents collaborate when trying to find a diagnosis when required.

In this paper we present our approach to this support system. It consists of a multi-agent system of case-based agents. Thus, each agent supports the diagnosis of a given hospital according to their experiences and criteria. However, all agents collaborate in order to achieve a clinical category for acute stroke diagnosis.

[1] University of Girona, Spain, email:blopez@eia.udg.es
[2] University of Girona, Spain, email:carles@eia.udg.es
[3] Hospital Dr. Josep Trueta, Spain, email:jserenal@meditex.es
[4] Hospital Dr. Josep Trueta, Spain

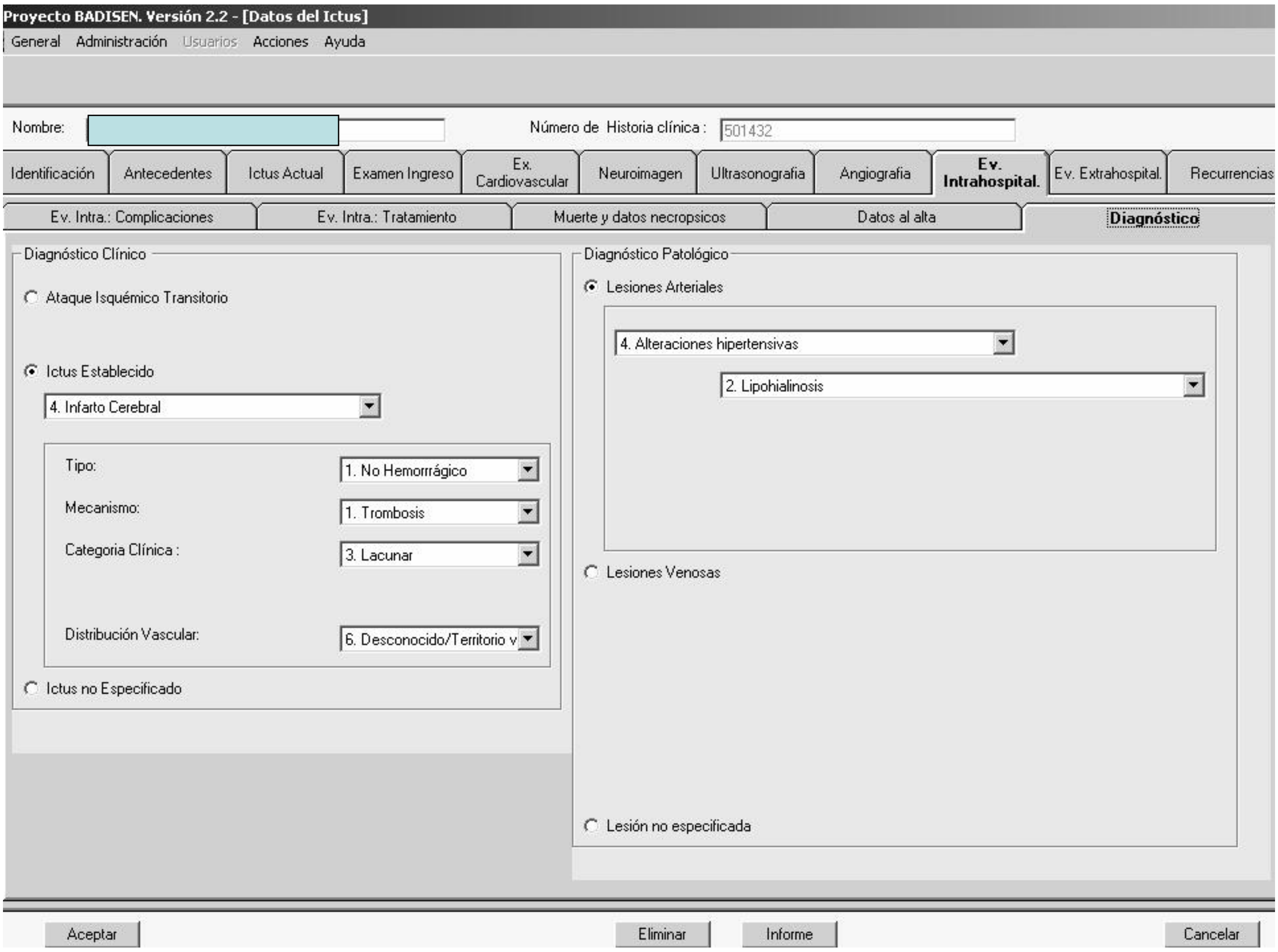

Figure 1. The user interface of the Badisen database

The paper is organized as follows. In Section 2 we present some related work. We continue in Section 3 by describing our multi-agent system of collaborative case-based agents. In Section 4, the decision making process of a single agent is provided. We proceed in Section 5 by providing some implementation hints, and we end in Section 6 with some conclusions.

2 RELATED WORK

There are several works related to multi-agent systems combined with case-based reasoning (multi-agent case-based reasoning). Two main approaches are generally followed. The first one assumes that each agent has a local view of the problem, so that it answers queries from other agents as required [15]. There is no single agent that has all the information. The second approach supposes that each agent is able to solve problems completely but is specialized in an area of expertise, that is, each agent has a different set of experiences. Then, cooperation can be focused either on sharing experiences as in [14, 10] or sharing solutions as in [12, 11].

When sharing experiences [14, 10] an agent can send cases to other agents to be solved by them. When sharing solutions, as in [12], each agent solves the problem at hand individually and makes its individual prediction, then all the predictions are added together to obtain an overall prediction. Our proposal follows the shared experiences approach of previous works. Among them, we highlight the work of Leake and Sooriarmurthi [7, 8] which deals with distributed case-bases that can reflect different user preferences. In this line, their work is closer to our situation in which all agents should be able to solve problems completely but the same problem can be solved differently by each agent according to the particular criteria the agent has (depending on the physician which uses the system).

3 COOPERATIVE CBR

Our multi-agent system is organized according to the different hospitals in a given zone (see Figure 2). Currently, four hospitals are represented by a different case-based agent (CB-agent) in the architecture. Then each CB-agent asses the diagnosis process according to the criteria of the physicians in a given hospital, and cooperates with other CB-agents when the assessment provided is not significant.

Thus, each agent is able to solve problems completely. The main difference between the agents is the criteria used to make the diagnosis, which mainly depends on the experience the neurologist has in using new technology.

Since the physician's criteria determines the diagnosis, the collaboration strategy of the multi-agent system consists in case exchanges instead of sharing solutions. That is, a CB-agent is not interested in the diagnosis of other CB-agents, but in the information from previ-

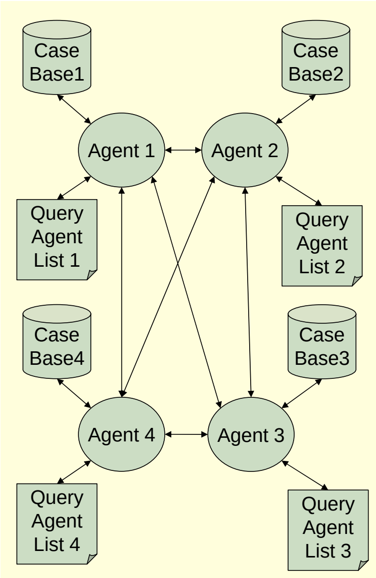

Figure 2. The multi-agent system

ous, similar cases. Then, if another agent has relevant cases to solve the problem, the agent receives these cases and will solve the problem from the cases retrieved from other agents. In this sense, this approach is close to the one proposed in [10].

In the rest of this section we describe the coordination protocol used for agent collaboration and the trust mechanism involved.

3.1 Coordination protocol

There is no central agent responsible for coordination, rather it is established in a peer-to-peer basis. Each agent has a list of agents in the multi-agent system, which is sorted according to how much it trusts the agents. Trust is related to the previous interactions of the system and computed according to [13], which has been applied previously to medical services [9]. When an agent requires help from other agents, he selects the first (most trusted) agent on the list a_i to broadcast the problem description. Then, if agent a_i has some similar cases, it gives the retrieved cases to the calling agent, together with the degree of similarity computed.

Figure 3 shows a state diagram of the coordination protocol between two agents. Circles are states, double circles are the initial state, and squares are final states. Then, from the initial state 0, the agent initiates the protocol, and localizes a possible partner in its list of agents. From state 1, the request is send from querying agent a to requested agent b in the message "send-problem". Then, querying agent a waits in state 2 until an answer is provided by requested agent b. If no answer is provided, agent a ends with a failure (state 4). Otherwise, the agent successfully ends in state 3.

In the case of failure, the agent tries with the next agent on the list. It is also possible that the received cases have a low similarity degree. Then, the protocol is started again with the next agent on the list.

Finally, if the cases received by the agent are used in the diagnosis process, the trust of the queried agent is increased as explained in the following section. Conversely (when the cases have not been useful), the trust is decreased.

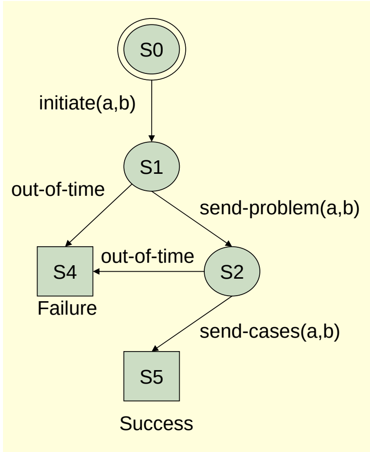

Figure 3. Coordination protocol for the peer to peer collaboration

3.2 Trust mechanism

Trust is defined as the agent's beliefs about attributes such as the reliability and honesty of the other agents with which it has interacted [18]. The trust an agent, a_i, has in an agent, a_j, is the accumulated evaluations that a_i has about a_j from past interactions. Once an agent a_i has interactions with agent a_j, its trust in agent a_j can be developed according to the degree of satisfaction it has with the interactions and this trust can be used to make decisions for future interactions.

In our domain problem, trust is used by each case-based agent to evaluate the reliability of the other agents based on the past history of satisfying and not satisfying the problems provided. This satisfying degree is related to the common criteria used by both agents to solve problems. So each agent a_i has a trust value for every other agent in the multi-agent system, forming its query agent list q_i:

$$q_i =< (a_{i_1}, t(i, i_1)), , (a_{i_n} t(i, i_n)) > \quad (1)$$

where n is the total number of agents in our system (n=4), a_{i_j} is an agent, and $t(i, i_j)$ represents the trust of agent a_i in agent a_j . Since each agent represents one hospital, note that it is assumed that all physicians of a given hospital (agent) follow the same decision criteria, and so have the same trust in physicians of another hospital. This assumption is reasonable if we take into account that in each hospital the head neurologist decides on the criteria with their colleagues.

A trust value t_j is defined in the [0,1] interval: 0 indicates an untrustworthy agent, while 1 indicates blind reliability. Trust between two agents, a_i and a_j , is then computed as follows [13]:

$$t(i, j) = E[B_j|\alpha, \beta] \quad (2)$$

where B_j is the variable that measures the probability that agent a_j fulfils its obligations, and E its expected value given the parameters α and β. According to [13], E is computed as:

$$E[B_j|\alpha, \beta] = \frac{\alpha}{\alpha + \beta} \quad (3)$$

Parameters α and β are defined, according to the authors, as the number of satisfying and unsatisfying services respectively, given a time window t. For our purpose, we assume an infinite time window. Then, the trust list is sorted in descending order, so that the first element corresponds to the agent with the highest trust.

This trust mechanism has been previously applied in a medical application successfully [9].

4 CASE-BASED AGENT

At the agent level, each agent relies on a Case-Based Reasoning (CBR) approach for decision making. CBR is applied with two different goals: when trying to find a diagnosis, and when trying to provide cases to another agent. In the first case, the four steps of the CBR cycle according to [1] have been extended to include using cases provided by other agents (see Figure 4). In the second case, a single retrieval step is enough.

Another particularity of our approach is the case structure. We are not dealing with flat cases, but with tree-structured cases. In the rest of this section, we describe the case-base and both CBR cycles (when making a decision, and when providing cases to neighbour agents).

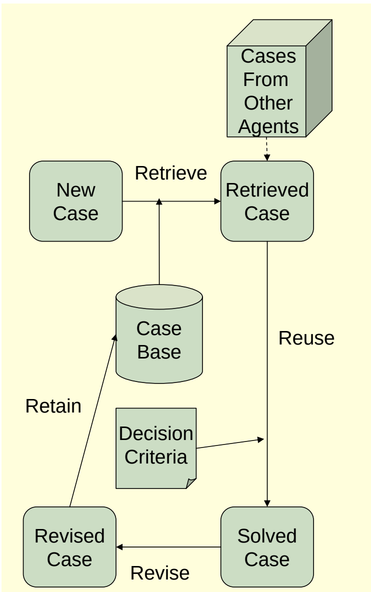

Figure 4. The different phases of the CBR cycle

4.1 The case structure and the case-base

A case represents all the information currently available in the BADISEN database [3] (see Figure 5). The following eleven data blocks were differentiated:

1. Identification data
2. History
3. Stroke data
4. Neurological examination results upon hospital admission
5. Cardiovascular exploration and laboratory data upon hospital admission
6. Neuroimage study
7. Ultrasonography
8. EchoCardio
9. Intra-hospital evolution
10. Extra-hospital evolution

All this data can be organized in a case by distinguishing two main parts:

1. Problem description: from block 1 to 8
2. Problem solution: blocks 9 and 10.

A case was formally defined as follows:

$$C_i =< D_i, S_i > \tag{4}$$

where D_i is the problem description, and S_i the problem solution.

4.1.1 *Problem description*

The problem description is made up of nine parts. There are several attributes in each part, which can be seen in Figure 5. The key issue in the data is the fact that not all attributes are present in all cases. For example, in the history of the patient data (issue 2), the cardiopathy attribute can have three values: yes, no and unknown. Only in the former situation, can the case contain the attributes cardiac arrythmia, ischemic arrythmia or embolic valvulopathy. We therefore consider the case structure of a case as a tree (acyclic graph).

Consistently, the notation used to represent the attributes is a set of nested attribute-value pairs. That is, D_i is defined as follows:

$$D_i = \{(x_j, v_j)\} \tag{5}$$

where $x_j \in X$, X is the set of attributes used to describe the problem, and v_j is the value of x_j which can be a list $\{(x_k, v_k)\}$ or a single value.

It is important to note that all the attributes have been coded as integers. However, they cannot be considered as such in order to perform mathematical operations with them, since they express discrete information (that is, they are codes). The 99 value corresponds to the unknown value, an [unfortunately] frequent value.

4.1.2 *Problem solution*

As the problem solution we are interested in the clinical diagnosis that belongs to the 9th data block of the BADISEN database (intra-hospital evolution data). The clinical category, as explained in the introduction, is determined in a two step process by expert neurologists by:

1. Applying a classification (TOAST [6] in our case).
2. Re-considering the result of the classification due to using additional technology (RM, ECO, ...).

The first step is easily performed by a new physician. The second step, if the solution is supported from an automatic system based on experiences as ours is, requires additional accompanying information so the physician can understand it. The information provided to the end user includes the differences between a current patient and a case in the memory. However, these differences are computed in the reuse step of the CBR cycle, and are not stored as part of the problem solution.

Therefore, the problem solution of the case that we store consists in the clinical category. Formally,

$$S_i = (clinicalCategory, v_j) \tag{6}$$

where v_j is one of the five possible values of clinicalCategory (*atherothrombothic, cardioembolic, small vessel disease , other, undefined*).

Identification data
Hospital code
Clinical history number
Name and surname
Address and phone number
City
Birth date
Gender
Point of departure
History
Familiar cases
Alcohol
Smoking
Arterial hypertension
Diabetes mellitus
Dyslipidaemia
Cardiopathy

  Cardiac arrythmia
  Ischemic cardiopathy
  Embolic Valvulopathy

Other cardiovascular diseases
Previous stroke

  AIT
  Transitory amaurosis
  Acute stroke
  Hemorrhagic stroke
  AVC undetermined

Previous treatment

  ...

Stroke data
Starting date and hour
Arrival at the UCIAs: date and hour
Admission data (Stroke unit)
Initial interval - neurological care
Symptom duration
Causing factors
How it was established
Initial moment
Accompanying symptoms

  Loss of consciousness
  Migraine
  ...

Previous Ranking scale
Neurological examination results upon hospital admission...
Cardiovascular exploration and laboratory data upon hospital admission ...
Neuroimage study ...
Ultrasonography ...
EchoCardio ...
Intra-hospital evolution ...
Extra-hospital evolution...

Figure 5. Attributes of a case (partial list).

4.2 CBR for decision making

In this section we describe the approach of the retrieve, reuse, revise and retain phases so the agent makes a decision with the collaboration of cases from other agents.

4.2.1 Retrieve

Given a current (new) case to be diagnosed, the retrieve phase recovers the most similar cases from the case-base. This phase of the cycle consists in two steps:

1. Matching the current case against all cases in the memory
2. Selecting the most similar cases

We have extended this classical view of the CBR cycle to include collaboration with other agents in the environment. Then, the following steps make up the retrieve phase:

1. Matching the current case against all cases in the memory
2. If the most similar cases have a similarity degree less than a given threshold θ, then:
   (a) Let be $\theta' = \theta$
   (b) While there are agents to ask and $\theta' \geq \theta$ do,
      i. Ask the next agent for relevant cases
      ii. θ = highest similarity degree of the provided cases
3. Selecting the k most similar cases

The matching and selection steps are described below. Collecting relevant cases from other agents was described in the previous section (coordination protocol).

Matching step. We require a metric which determines how similar two cases are. According to our case structure, we need to compare two tree structures. For example, Figure 6 shows the structure of the new case at the top and at the bottom a possible case in the memory. Their corresponding representation is the following:

$$\begin{aligned} D_{new} = \quad & \{(a, v_a), \\ & (b, \{(b_1, v_{b1}), (b_2, v_{b2}), (b_3, v_{b3})\}), \\ & (c, v_c), \\ & (d, v_d)\} \end{aligned}$$

$$\begin{aligned} D_{mem} = \quad & \{(a, v_a), \\ & (b, v_b), \\ & (c, v_c), \\ & (d, \{(d_1, v_{d1}), (d_2, v_{d2})\})\} \end{aligned}$$

Then, our similarity metric is defined based on the similarity between two trees. It is defined as a weighted average as follows:

$$sim(D_{new}, D_{mem}) = \frac{\sum_{i \in D_{new}} \omega_i sim_i(v_i^{new}, v_i^{mem})}{\sum_{i \in D_{new}} \omega_i} \tag{7}$$

where ω_i is the weight expressing the relevance of the i attribute, and v_i^{new} and v_i^{mem} are the values of the i attribute in the new and memory case correspondingly, and $sim_i(v_i^{new}, v_i^{mem})$ the similarity between these values. Note that if attribute i is not present in the memory case, this function is assumed to be 0.

Given an attribute x_i, the similarity of two of their values is computed as follows:

$$sim_i(v_j, v_k) = \begin{cases} 0 & \text{if } v_j \text{ is a single value and } v_k \text{ is a tree structure} \\ 0 & \text{if } v_j \text{ is a tree structure and } v_k \text{ is a single value} \\ 1 - \delta(v_j, v_k) & \text{if } v_j \text{ and } v_k \text{ are single values} \\ sim(v_j, v_k) & \text{otherwise} \end{cases} \quad (8)$$

where δ is the Hamming distance of two values, and $sim(v_j, v_k)$ is the similarity between two trees. On one hand, δ is computed as follows:

$$\delta(v_j, v_k) = \begin{cases} 1, & \text{if } v_j \neq v_k; \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

On the other hand, note that using $sim(v_j, v_k)$ inside the definition of $sim_i(v_j, v_k)$ makes our similarity measure recursive.

Trying to provide the attribute relevance ω_i manually is a hard issue in any case-based system. In order to facilitate its acquisition, we propose defining nine attribute relevances, one for each part of the problem description.

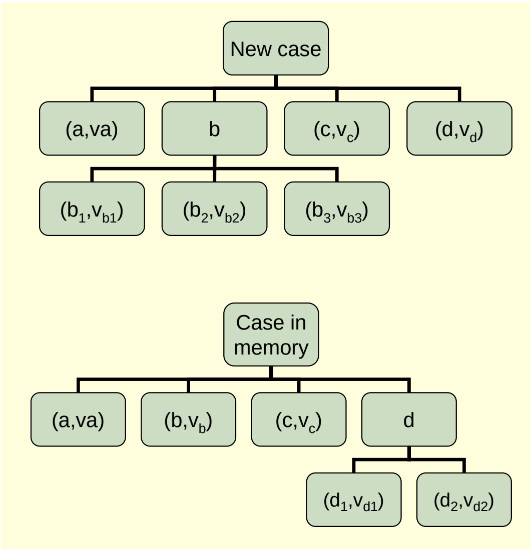

Figure 6. Two different cases for matching

Finally, we stress the fact that the metrics presented in this section are a first approximation to the problem. Further experimentation on them following [17] and [16] will probably improve our expectations.

Selection step. Once all the similarities of the current case with the memory cases have been computed, the selection step consists in choosing the most similar cases. We considered the k-most similar (a k-neighbour approach). We believe that with k=5, enough information is provided to the physician.

Note that we always get some cases either from the memory or from other agents. In the best case, the retrieved cases are over the threshold θ.

4.2.2 *Reuse*

In the reuse phase the k-most similar cases are used to elaborate the solution to the current case. This is the hardest phase of the CBR approach. In our medical application, we aim to facilitate the physician with guideline information that helps him/her to find the appropriate diagnosis. Thus, the solution of the new case consists in a list of the k-most similar cases, sorted according to their similarity degree. For each case, the following information is provided as the solution:

- The clinical category
- The differences between the current case and the case in the memory

With this information, the physician can, for example, either make an additional analysis of the patient or revise the data about him/her in order to definitively check if the current patient is suffering from the same illness as the previous patient.

4.2.3 *Revise*

In the revise phase, the case-based agent learns about the problem solution, whenever it has been successful or not. For evaluating this, it requires some feedback from the user. We gather two kinds of feedback which is explicitly provided:

- The final clinical category
- Whether the reported cases have been useful or not.

The feedback is used to both retain the case and to modify the trust of a neighbourhood agent in the case that this agent has provided the cases used to determine the diagnosis (see Section 3.2).

4.2.4 *Retain*

All cases are retained in the memory because accumulating experiences is a key issue in medical practice in order to elaborate further statistical analysis. Therefore, there is no special retention policy, and we always learn from the solved cases. In the best situation, the case is complete; while in the worst one, the case has the clinical category of the stroke empty.

4.3 CBR for providing cases

Given a (new) problem case provided by a neighbourhood agent, the case-based agent tries to find similar cases with which to answer the query by following the retrieve phase. In this case, we follow the classical approach:

1. Matching the current case against all cases in the memory
2. Selecting the most similar cases

The methods described in the previous section for matching and selecting are the same as these ones.

5 A CASE EXAMPLE

In this section we illustrate our methodology with two examples. In the first example we assume a similarity threshold of $\theta = 0.6$, while in the second example θ is set to 0.7.

5.1 Example with $\theta = 0.6$

Suppose that agent $A1$ is trying to determine the diagnosis of a patient C_{new}. Then, it applies the CBR cycle in order to make its own decision. From its own experience (its own case-base), agent $A1$ recovers case C_{23} with a similarity value of 0.65, and other cases with similarity degree under 0.3. As the similarity threshold is $\theta = 0.6$, case C_{23} is the result of the retrieve phase.

The diagnosis of C_{23} has been labelled with the *undefined* clinical category. Therefore, few insights can be taken from case C_{23} to solve the new case.

5.2 Example with $\theta = 0.7$

In this new situation, the similarity degree of 0.65 of case C_{23} is not enough to satisfy the retrieval criteria. So agent $A1$ queries the other agents about previous experiences. We assume that agent $A1$ has the following list of agents:

$$Q1 =< (A3, 0.9), (A2, 0.7), (A4, 0.4) >$$

Then, the first agent $A3$ is selected in order to ask it about past experiences. Agent $A3$ uses its case-base to recover past cases from the memory. It recovers case C_{54} with a similarity degree of 0.8 ($\geq \theta$). This case is returned to agent $A1$.

On receiving case C_{54}, agent $A1$ continues with the reuse phase. Case C_{54} supports the clinical category of $atherothrombothic$. One of the differences is that in the current case C_{new} the ECO image has not been obtained, in addition to $diabetes$ and $arteriosclerosis$ attributes being present in the memory case (from $antecedents$). Then, the solution prompts the user as shown in figure 7.

CASE $C54$
CLINICAL DIAGNOSIS: ACUTE STROKE
Clinical category: atherothrombothic
MAIN DIFFERENCES WITH THE CURRENT CASE:

- ECO: empty.
- ANTECEDENTS: sclerosis, diabetes

Figure 7. Solution provided in the second example

In the revise phase, the physician gives the system the feedback that the case has been useful for solving the current diagnosis. Then, the case is retained in the memory. Additionally, the trust in agent $A3$ is increased from 0.9 to 0.93 according to Equation 2.

5.3 Discussion

Using a low similarity threshold allows the agent to recover cases from its own experience easily. However, the diagnosis can be enriched if the similarity threshold is increased and other agents participate in the diagnosis. In some ways, the other agents are assessing the physician's decision making process, as may happen in real life.

6 IMPLEMENTATION ISSUES

We are currently in the process of implementing a prototype of the architecture. We are studying the benefits of using the Noos kit [2, 14] to develop our system since this development environment provides two key issues regarding our problem:

- A multi-agent platform for case-based agents
- The Noos language to define the case-based agents

In order to test the system, we are using 935 cases collected from 4 hospitals thanks to previous works of the Spanish Neurological Society in BADISEN [3].

7 CONCLUSIONS

In this paper we have presented a collaborative multi-agent approach based on case-based agents that support acute stroke diagnosis. The multi-agent system is organized according to the different hospitals involved in this kind of illness and assumes that each hospital follows different decision criteria when determining the diagnosis due to different experiences in using new technologies. Therefore, each agent keeps information about the experiences in acute stroke diagnosis and their corresponding patient data.

When a new case is presented, an agent tries to find a diagnosis according to its past experience. If it is not able to provide a confident diagnosis, it asks for the collaboration of other agents in the system according to a peer to peer mechanism and a trust relationship.

In the near future, we expect to have the first prototype of the system running in a Noos platform. The data in BADISEN is our workbench for testing our system.

ACKNOWLEDGEMENTS

This research project has been partially funded by the Spanish MEC project TIN2004-06354-C02-02 and the DURSI-AGAUR 00296 SGR (2005).

REFERENCES

[1] Agnar Aamodt and Enric Plaza, 'Case-based reasoning: Foundational issues, methodological variations, and system approaches', *Artificial Intelligence Communications*, **7**(1), 39–59, (1994).

[2] Josep Lluis Arcos, *The NOOS representation language*, Phd, Technical University of Catalunya, 1997.

[3] Badisen. http://www.ictussen.orgmedicosServiciosServicios_BADISEN.htm.

[4] J Bogousslavsky, G Van Melle, and F Regli, 'The lausanne stroke registry: analysis of 1,000 consecutive patients with first stroke', *Stroke*, **19**, 1083–1092, (1988).

[5] M. Fisher and K. Takano, 'The penumbra, therapeutic time window and acute ischaemic stroke', *Bailliees Clin Neurol*, **4**(2), 279–295, (1995).

[6] Peter L. Kolominsky-Rabas, Margarete Weber, Olaf Gefeller, Bernhard Neundoerfer, and Peter U. Heuschmann, 'Epidemiology of ischemic stroke subtypes according to toast criteria incidence, recurrence, and long-term survival in ischemic stroke subtypes: A population-based study', *Stroke*, **32**, 2735, (2001).

[7] David B. Leake and Raja Sooriamurthi, 'When two case bases are better than one: Exploiting multiple case bases', in *ICCBR*, pp. 321–335, (2001).

[8] David B. Leake and Raja Sooriamurthi, 'Managing multiple case bases: Dimensions and issues', in *FLAIRS*, pp. 106–110, (2002).

[9] Beatriz Lopez, Silvana Aciar, Bianca Innocenti, and Isabel Cuevas, 'How multi-agent systems suport acute stroke emergency treatment', in *IJCAI Workshop on Agents Applied in Health Care*, pp. 51–59, (2005).

[10] Lorraine McGinty and Barry Smyth, 'Collaborative case-based reasoning: Applications in personalized route planning', in *ICCBR*, pp. 362–376, (2001).

[11] Miquel Montaner, Beatriz Lpez, and Josep Lluis de la Rosa, 'Opinion-based filtering through trust', in *Cooperative Information Agnets (CIA)*, p. 164, (2002).

[12] Santi Ontanon, *Ensemble Case Based Learning for Multi-Agent Systems*, Phd, Technical University of Catalonia, 2005.

[13] Teacy W. T. L. Jennings N. R. Patel, J. and M. Luck, 'A probabilistic trust model for handling inaccurate reputation sources', in *Proceedings of Third International Conference on Trust Management*, (2005).

[14] Enric Plaza, Josep Lluis Arcos, and Francisco Martin, *Cooperative case-based reasoning*, 180–201, In Gerhard Weiss, editor, Distributed Artificial Intelligence Meets Machine Learning. Learning in Multi-Agent Environments, Lecture Notes in Artificial Intelligence, 1221, Springer-Verlag, 1997.

[15] M.V. Nagendra Prassad, Victor R. Lesser, and Susan Lander, 'Retrieval and reasoning in distributed case bases', Technical report, U. Mass, Computer Science Department, (1995).

[16] Hector Nunez Rocha, *Feature Weighting in Plain Case-Based Diagnosis*, Phd, Technical University of Catalonia, 2004.
[17] D. Randall Wilson and Tony R. Martinez, 'Improved heterogeneous distance functions', *Journal of Artificial Intelligence Research*, **6**, 1–34, (1997).
[18] Bin Yu and Munindar P. Singh, 'To-wards a probabilistic model of distributed reputation management', in *4th Workshop on Deception, Fraud and Trust in Agent Societies*, (2002).