Auction-based Retrieval

Francisco J. Martin and Enric Plaza
IIIA - Artificial Intelligence Research Institute
CSIC - Spanish Council for Scientific Research
Campus UAB, 08193 Bellaterra, Catalonia, Spain.
Vox: +34-93-5809570, Fax: +34-93-5809661
{martin,enric}@iiia.csic.es

Abstract

We present Auction-based Retrieval (ABR) an approach for distributed case retrieval based on the economic metaphor of *auction* and on our current research on agent-based electronic trading. We focus on agent-mediated systems where each agent is able to reason from a (privately owned) case-base, has own interests, and nonetheless it is able to cooperate with the other to solve new problems. In this situation (called CoopCBR[13]) case retrieval has an added difficulty, namely the coordination of case retrieval processes from multiple case-bases. Market institutions like auctions offer precisely well-known and well-founded coordination mechanisms for situations were the participants have limited information and individual objectives and interests.

Keywords: Multi-Agent Systems, Interagents, Communication, Coordination

1 Introduction

As more CBR systems are applied to the internet, the issues of agenthood and multiagent coordination and cooperation will become increasingly important for case-based reasoning. We present in this paper a method for case retrieval in a multiagent setting based on the economic metaphor of *auction*. We are interested in societies where agents are autonomous and, although amenable to cooperate, they pursue their own interests. In this setting, agents may pertain to different institutions and are in charge of maintaining and securing information that is not generally accessible. Agents capable of performing case-based reasoning would then need transparent and certified protocols to cooperate with other CBR agents dealing with the same application domain but residing elsewhere and pertaining to a different institution. In particular, case retrieval among a society of CBR agents is a kind of task that will require a transparent and certified protocol if it is to be effectively deployed. Auction houses are institutions that provide such transparency and certification by means of a set of public rules an protocols that are public and enforcible. A protocol for case retrieval based on the auction metaphor inherits the advantages of transparency, certification, and recognizes the agents features of autonomy, discretionary action and self-interest.

In the rest of the section we will introduce our multiagent setting for CBR agents, where we explain the assumptions that we require from the agents, and later we summarize the relevance of auctions in networked information retrieval; the section closes with an introduction of retrieval patterns, the elements of the retrieval process to be used later on the paper to define auctions and the protocol of auction-based retrieval.

1.1 The CoopCBR Framework

In what follows we shall assume that we are dealing with multi-agent systems that fulfill the following assumptions.

Homogeneous Agents The representation languages of the involved agents are the same. Consequently, communication among agents does not require a translation phase.

Peer Agents The involved agents are capable of solving the task in hand. In other words, cooperating agents are not merely specialists at specific subtasks. Instead, they are capable of solving the overall task by themselves (most of the time, at least).

Learning Agents The agents solve the task based knowledge acquired by learning from their individ-

ual, usually divergent, experience in solving problems and cooperating with other agents in solving problems.

We call this framework CoopCBR [13, 7] because it is applied to co-operative case based reasoning architectures. In order to enable the agents to share their experience for the task of retrieving adequate cases, two "co-operation modes" have been developed. In the DistCBR mode, an agent delegates the task of finding an adequate case to an acquaintance, which selects a case based on its local preference criteria. In the ColCBR mode the requesting agents additionally provides the preference criteria for remote case selection in the shape of a mobile method.

1.2 Auctions for Networked Information Retrieval

Auctions are an attractive domain of interest for AI researchers in at least two areas of activity. On the one hand, we observe that the proliferation of online auctions in the Internet —such as Auctionline, Onsale, InterAUCTION, eBay and many others— has established auctioning as a main-stream form of electronic commerce. Thus, agent-mediated auctions appear as a convenient mechanism for automated trading, due mainly to the simplicity of their conventions for interaction when multi-party negotiations are involved, but also to the fact that on-line auctions may successfully reduce storage, delivery or clearing house costs in many markets. This popularity has spawned AI research and development in auction servers[22, 17] as well as in trading agents and heuristics[5, 9]. On the other hand, auctions are not only employed in web-based trading, but also as one of the most prevalent coordination mechanisms for agent-mediated resource allocation problems (f.i. energy management[23], climate control[6], flow problems[20]).

For instance, the contract net protocol [18, 4] facilitates distributed control of cooperative task distribution (task-sharing) among the nodes of a distributed problem solver. In this approach, when a particular node lacks necessary knowledge to perform a task it announces (broadcasts) that task to the rest of nodes. Then, each node evaluates its own suitability to carry out that task and bids a figurative price (since each node is endowed with a strength parameter as the hypothetical capital for granting contracts or charging services) for that task. Afterwards, the originator chooses the most suitable among all bids submitted. A shortcoming of the contract net protocol is the communication overhead caused by broadcasting. Recent approaches—like addressee learning based on CBR [10]—have tackled this problem.

In this paper we present ideas and tools that could be relevant to networked information retrieval in general and to distributed case retrieval in particular.

The bidding mechanism that we propose in this work is based on DBP auctions[15]. In these auctions, basically, the auctioneer opens a new *bidding round* to auction a good among a group of agents, he starts quoting offers downward from the chosen good's starting price. For each price called, three situations might arise during the open round: i) several buyers submit their bids at the current price. In this case, a collision comes about, the good is not sold to any buyer, and the auctioneer restarts the round at a higher price; ii) only one buyer submits a bid at the current price. The good is sold to this buyer whenever his credit can support his bid. Whenever there is an unsupported bid the round is restarted by the auctioneer at a higher price, the unsuccessful bidder is punished with a fine, and he is expelled out from the auction room; and iii) no buyer submits a bid at the current price. If the reserve price has not been reached yet, the auctioneer quotes a new price which is obtained by decreasing the current price according to the price step. If the reserve price is reached, the auctioneer declares the good *withdrawn* and closes the round.

1.3 Modularizing Retrieval through Patterns

Our goal is to construct a distributed, multi-agent case retrieval process based on the bidding mechanism explained above. A main difference in CoopCBR from the single CBR agent is that now cases are not only distributed but they are owned by several agents. In general, cases belonging to different agents may have several properties that have to be enforced (like maintaining privacy, insuring property rights, etc). The auction metaphor provides a regulated environment where these properties are taken into account (see next section), but first we need to determine how the retrieval characteristics can fit into this coordination mechanism.

A process of case retrieval, in abstract terms, selects one case from a collection of cases[1]. Similar-

[1] We can also consider a two step approach to retrieval.

ity measures are a device to estimate the degree of relevance of precedent cases with respect to the current problem case. From an abstract stance similarity measures are useful because they provides a (partial) ordering over the cases in a case base. If the estimation is good, taking the highest case in the ordering the CBR system can lead to a correct solution. Similarity metrics are often used for tuple-based representations of cases (i.e. cases are represented by attribute value vectors). In essence, what we need to perform case retrieval is *a problem-centered partial ordering of a collection of cases.*

Similarity measures are not the only way to induce partial ordering onto a case base. Indeed, establishing similarity measures based on metrics is difficult for structured cases. A structured representation of cases is one where case t_i is such that $t_i.a_j = t_k$, i.e. where the value of an attribute of a tuple (case) is another tuple t_k. The notion of similarity in structured cases is then the notion of shared structure, and this notion can be formalized by means of the antiunification concept[12]: given a subsumption order ($\sqsubseteq$) antiunification $AU(a,b) = z$ provides a term that subsumes both ($z \sqsubseteq a \wedge z \sqsubseteq b$) and does not subsume another term that also subsumes a and b.

When domain knowledge is available, we can construct *patterns* expressing the most important features of a problem; using this patterns we can extract from the current problem that substructure pertaining to the pattern. For instance, Perspectives in the Noos knowledge representation language[1] is a mechanism for using a declarative representation of domain knowledge in case retrieval. In Noos, given a current problem c and a syntactic pattern π (encoding the relevant features) a perspective method P constructs a perspective on the current case $P(\pi, c) = c_\pi$, i.e. constructs an instantiation of π with the values of the current problem c. We call c_π the π-perspective of c. Later, c_π can be used as a retrieval pattern (or "query") to the case base CB, retrieving all cases that satisfy it: $\{c_i \in CB | c_\pi \sqsubseteq c_i\}$.

We will adapt these concepts to a multi-agent setting as follows. The CBR agent that wants to perform case retrieval, for a specific current problem c, has an ordered sequence of patterns $\Pi = (\pi^1 \ldots \pi^n)$ where each π^{i+1} is not more restrictive than the previous π^i ($\pi^{i+1} \sqsubseteq \pi^i$ or ($\pi^{i+1} \not\sqsubseteq \pi^i \wedge \pi^i \not\sqsubseteq \pi^{i+1}$)). From Π a CBR agent uses perspective method to generate, for a current problem c, an ordered sequence of retrieval patterns $Q = (c_{\pi^1} \ldots c_{\pi^n})$. Since Q is also ordered, the CBR agent can now ask the first query c_{π^1} to other agents to see whether they have any case that satisfies it. If not, the CBR agent can ask the next (less or equally restrictive) query to other agents. Following this mechanism, the CBR agent will receive the best case with respect to the order established by itself in the sequence Π. In fact, this mechanism introduces a partial order on the collection of cases owned by several agents in a multi-agent setting.

Finally, privacy and ownership properties are controlled by the CBR agent. Since the CBR does not need to send the whole case to other agents to use it for their own retrieval methods privacy and ownership are insured. The CBR agent is free to use the collection of patterns Π best suited for its purposes, and sensitive information can thus be hidden. Moreover, we can view the sequence of queries Q as generic descriptions of the situations in which the CBR is interested: the CBR does not send the case over the network of agents. As we will show presently, a CBR agent sends a "contract proposal" and other agents can "bid" if they know about a situation that satisfies that description. If the case the other agent sends to the CBR agent does in fact meet the requirements then the contract is granted and the sender receives the corresponding fee.

The remainder of this work is organized as follows. In the next Section we explain Auction-based Retrieval (ABR) in depth. In Section 3 we will describe the Plural platform that has allowed us to apply ABR in CoDiT—an agent-based system for therapy recommendation in diabetic patients. Finally, Section 4 provides some discussion about ABR.

2 Auction-Based Retrieval

In our approach, when a CBR agent a is not capable of finding a "good" precedent on its own case-base, it will summon an *auction* in order to find an appropriate case owned by other agent(s). Thus, agent a will select the *participant* agents (that will play the role of *bidders*) and will conduct the auction process (playing the role of *auctioneer*). The *good* to be auctioned is the (hypothetical) benefit

Sometimes a small set of ranked cases is said to be retrieved from the case-base. However, one of this cases has to be selected in a second phase of retrieval. Often the CBR system can backtrack to other cases of the small set if the first ones were no good. From an abstract stance they are varieties of the generic "selection of one case" process.

obtained for finding a "good" precedent. A CBR agent can use the same auction to sell more than one good. In this case, each good is auctioned in a different *round*.

For the purpose of identifying and characterizing the ontological elements involved in an auction, we define, following [16], an *auction descriptor* as follows:

Auction Descriptor We define an Auction Descriptor $\mathcal{A}^{a,n}$ for the nth auction of auctioneer agent a as the 3-tuple $\langle \mathcal{B}_n^a, \mathcal{G}_n^a, \mathcal{D}_n^a \rangle$:

- $\mathcal{B}^{a,n}$ is a non-empty, finite set of bidders' identifiers representing the set of participants in auction nth of agent a.
- $\mathcal{G}^{a,n}$ is a non-empty, finite set of goods' identifiers representing the set of goods to be auctioned in auction nth of agent a. Auction nth of agent a will have a round for each good in $\mathcal{G}_n^a$.
- $\mathcal{D}^{a,n}$ stands for the instance of the particular *bidding protocol dynamics descriptor* to be deployed in auction $\mathcal{A}^{a,n}$. A bidding protocol dynamics descriptor defines the different parameters followed by a CBR agent in an auction guided by a particular bidding protocol as we will see further on.

For each good $g_i^{a,n}$ in $\mathcal{G}^{a,n}$, agent a will generate a collection $R_i^{a,n}$ of *offers*.

Offer We define an offer as a pair $\langle c_{\pi^i}, p \rangle$ where c_{π^i} is a pattern for retrieval and p is a real number that represents the price that agent a is disposed to pay for finding a precedent case subsumed by retrieval pattern c_{π^i}.

A collection $R_i^{a,n}$ of offers will be partially ordered according to agent a's preferences about offer patterns $Q = (c_{\pi^1} \ldots c_{\pi^n})$ and totally ordered with respect to offer prices p. Moreover, each good has associated a *starting offer* p_α the first pair $\langle P, p \rangle$ to be offered and a *reserve offer* p_ω the offer at which the good will not be sold and therefore withdrawn, the last offer generated. Notice that p_α represents the most specific pattern and therefore the highest offer whereas p_ω, on the contrary, represents the most general pattern and therefore the lowest offer.

In an auction $\mathcal{A}^{a,n}\langle \mathcal{B}_n^a, \mathcal{G}_n^a, \mathcal{D}_n^a \rangle$ each CBR agent $b_i^{a,n} \in \mathcal{B}_n^a$ acting as a *bidder* receives offers containing the corresponding retrieval pattern that it will use in its privately owned case-base CB_i to search for good precedent cases.

An agent $b_i^{a,n}$ that has a precedent case C_i^r in CB_i befitting offer r sends a bid to the auctioneer offering case C_i^r. Notice that it will only send a request for bid and not the actual case yet. Then, if the auctioneer accepts that bid it will request case C_i^r. Once the auctioneer checks the goodness of case C_i^r it will declare bidder $b_i^{a,n}$ the *winner*. Notice that whereas the auctioneer agent conducts most of the auction process: it selects which are the bidders, selects the offers to be sent and the order in which they are sent; and also takes charge of deciding whether the bidder offer is "good enough" for winning the round. The decision of when submit a bid is up to the bidder agent, which can decide to submit a bid only when it considers appropriate.

In an auction $\mathcal{A}^{a,n}\langle \mathcal{B}_n^a, \mathcal{G}_n^a, \mathcal{D}_n^a \rangle$ agent a keeps track of the *credit* that measures the credibility given to each bidder agent $b_i^{a,n} \in \mathcal{B}^{a,n}$.

Credit We define the credit of an agent i as the mapping $\mathcal{C}_i^{a,n} : \mathcal{B}^{a,n} \longrightarrow I\!R$ that assigns to each buyer in $\mathcal{B}^{a,n}$ his available credit during auction nth of agent a.

The auctioneer will either increase or decrease a bidder's credit according to the bids it submits (i.e. the goodness of the cases sent). When an agent loses its credit the auctioneer will not take into account its subsequent bids, will expel it from the current auction and probably will not summon it for further auctions.

Information Revelation The auctioneer also decides the policy followed for information revelation (bidders identity, bids, collisions, expulsions, winner's bid, winner identity, etc.). In our approach, we distinguish two kind of auctions: *public* and *private* according to the range of information made public by auctioneer agent. In public auctions, all participants receive information concerning all other participants while in private auctions a bidder will only receive information concerning itself. Notice that whereas private auctions can a priori diminish communications overhead, information revealed during public auctions can be acquired and used a posteriori by involved agents. That is, agents can later address directly subsequent problems and lessen, therefore, communication costs.

Auction Scheduling Once a CBR identifies the potential for calling for an auction. It will announce to all chosen participants indicating the starting

time and the auction descriptor (according to the character of the auction: public or private). Afterwards, interested bidders have to send a request for admission. In the same agent society can concur more than one auction, it is up to agents to establish starting times and take part in more than auction at a time.

When auctioning a good, one could choose among a wide range of bidding protocols (DBP,UBP[2], etc.). Each of these protocols can be characterized by a set of parameters that we refer to as *bidding protocol dynamics descriptors*, so that different instantiations of such descriptors lead to different behaviors of their corresponding bidding protocols. As a particular case, we will concentrate on the downward bidding protocol (DBP). Thus, we state explicitly the bidding protocol (as described in [16][3]) along with its respective parameterization.

2.0.1 Dutch Auction-based Retrieval

1. The auctioneer chooses a good out of a set of goods.

2. With a chosen good g, the auctioneer opens a *bidding round* by quoting offers (namely retrieval patterns for the current problem and the corresponding resale price) downward from the good's starting offer p_α containing one of the most restrictive patterns.

3. The auctioneer waits for bids, but only a preestablished time slot Δ_{offers}. For each offer called by the auctioneer, several situations might arise during the open round:

   No bids No buyer submits a bid at the current price. If the reserve offer (p_ω containing one of the least restrictive pattern has not been reached yet, the auctioneer quotes a new price. If the reserve offer is reached, the auctioneer declares the good *withdrawn* and closes the round. That is, the auctioneer has no more retrieval patterns and therefore the round is closed without any precedent case being retrieved.

   One bid Only one buyer submits a bid at the current price. The good is sold to this buyer whenever his credit can support his bid. Namely, the auctioneer asks the agent b_i to send its precedent case C_r^i such that $C_r^i \sqsubseteq p$. If later this case turns out to be no good, $C_r^i \not\sqsubseteq p$, i.e. there is an unsupported bid, the unsuccessful bidder is expelled out from the auction, and the round is restarted by the auctioneer at a higher offer.

   Multiple bids Several buyers submit their bids at the current offer. In this case, a collision comes about, the good is not sold to any buyer, and the auctioneer restarts the round at a higher offer. Nevertheless, the auctioneer tracks whether a given number of successive collisions (Σ_{coll}) is reached, in order to avoid an infinite collision loop. This loop is broken by randomly selecting one buyer out of the set of colliding bidders.

4. The first three steps repeat until the auctioneer has no more goods left.

Five parameters that control the dynamics of the bidding process are implicit in this protocol definition[4]. We shall enumerate them now, and require that they become previously instantiated by the auctioneer as part of an auction definition.

Dutch Dynamics Descriptor We define a Dutch Dynamics Descriptor $\mathcal{D}_{DBP}$ as the 5-tuple $\langle \Delta_{price}, \Delta_{offers}, \Delta_{rounds}, \Sigma_{coll}, \Pi_{sanction}, \Pi_{rebid} \rangle$ such that

- $\Delta_{price} \in I\!N$ (price step). Decrement of price between two consecutive quotations uttered by the auctioneer.
- $\Delta_{offers} \in I\!N$ (time between offers). Delay between consecutive price quotations.
- $\Delta_{rounds} \in I\!N$ (time between rounds). Delay between consecutive rounds belonging to the same auction.
- $\Sigma_{coll} \in I\!N$ (maximum number of successive collisions). This parameter prevents the algorithm from entering an infinite loop as explained above.

[2] DBP stands for *downward-bidding protocol* (or "Dutch") and UBP stands for *upward-bidding protocol* (or "English")

[3] http://www.iiia.csic.es/Projects/fishmarket

[4] Other bidding protocols—f.i. UBP, Yankee, Double auction, etc.—would be characterized by other sets of parameters)

- $\Pi_{rebid} \in I\!R$ (price increment). This value determines how the new offer is calculated by the auctioneer from the current offer when either a collision, or an unsupported bid occur.

Note that the identified parameters impose significant constraints on the trading environment. For instance, Δ_{offers} and Δ_{rounds} affect the agents' time-boundedness, and consequently the degree of situatedness viable for bidding strategies. Notice that buyers are expelled at the first unsupported bid that they submit . A sanction factor could be used to expel a buyer only when a number of unsupported bids is reached.

3 The Plural Platform for CBR agents

Auction-based Retrieval requires adequate communication and coordination mechanisms in order to ensure the fulfilment of the bidding protocol in use. Moreover these mechanism have to be flexible enough to support different bidding protocols so that the most efficient bidding mechanism can be set according to the domain application. Plural is a platform for sharing knowledge and experience among cognitive agents that co-operate within a distributed case-based reasoning framework [7]. Plural provides the infrastructure that allow CBR agents to follow the bidding protocols set by auctions. Plural is best explained in term of its four components:

Noos is a reflective object-centered representation language designed to support knowledge modeling of problem-solving and learning [2]. Noos is based on the task/method decomposition principle. For instance, Figure 1 shows the task/method decomposition of a case-based reasoning method for diabetes therapy. The `CBR-method` is decomposed into two subtasks, namely `retrieve` and `reuse`. For each subtask one or several alternative methods are specified—e.g. subtask `retrieve` has the `Identify&Select` method that is decomposed in (sub)tasks namely `identify`, `search`, and `select`.) and the analysis of knowledge requirements for methods. Noos is related to knowledge modeling frameworks like KADS [21] or components of expertise [19]. CBR agents' problem-solving capabilities are implemented on top of the Noos language.

Plural constructs Current knowledge representation languages have been thought to develop isolated applications and usually cannot provide such applications with the communication abilities required by agent-based systems. Plural constructs provide a seamless extension of Noos with agent-oriented capabilities [7]. Plural can be thought of as an extension of a knowledge representation language with an agent communication language and an agent conversation language. Both of these languages have been provided at the knowledge representation level by means of a set of constructs with the same nature as the rest of Noos constructs. The Plural extension to Noos determines the agents' social abilities. Based on facilities for sharing data and code over IP networks, task delegation and mobile methods are supported. In addition, each agent is provided with acquaintance models specifying the tasks and the problem types that the modeled partner can cope with [7].

Conversation Protocols We view conversations as the means of representing the conventions adopted by agents when interacting through the exchange of utterances[3] —"utterance suggests human speech or some analog to speech, in which the message between sender and addressee conveys information about the sender"[11]. More precisely, such conventions define the *legal* sequence of utterances that can be exchanged among the agents engaged in conversation: what can be said, to whom and when. Conversations in agent-based systems can be conceptualized, modeled and implemented using Conversation Protocols (CPs) [8]. *Conversation protocols* are coordination patterns that constrain the sequencing of utterances during a conversation. CPs are a special type of Pushdown Transducers (PDTs) that allows to store the context of ongoing conversations, and, in contrast with other approaches, that provides a mapping from specification to implementation. Moreover, as a distinctive feature from other approaches, CPs provide a detailed analysis that studies the properties that conversation protocols must exhibit in order to ensure protocol compatibility, and therefore the soundness of agent conversations [8].

Interagents are autonomous software agents that mediate the interaction between each agent and the agent society wherein it is situated [8]. Interagents employ conversation protocols for mediating conversations among agents. Interagents are responsible for posting utterances of its *customer*[5] to the corresponding addressee and for collecting the utterances that other agents address to its cus-

[5] We call customer the agent exploiting and benefiting from the services offered by an interagent

tomer. Each interagent has a collection of relevant conversation protocols (CP) used for managing its customer conversations. When its customer intends to start a new conversation with another agent the interagent instantiates the corresponding conversation protocol. Once the conversation starts, the interagent becomes responsible for ensuring that the exchange of utterances conforms to the CP specification. Before setting up any conversation the interagent must perform a *CP negotiation* process with the interagent of the addressee agent. The goal of CP negotiation is to reach an agreement with respect to the conversation protocol to be used. Moreover, before starting a conversation, the interagent performs a *CP verification* process. This process checks whether the CP to be used verifies the necessary conditions (liveliness, termination, deadlock and race condition free) for guaranteeing the correct evolution of an interaction. Finally, an interagent allows its customer to hold several conversations at the same time. This capability for *multiple conversations* is important because, although in the paper we consider only conversations with two participants (dialogues), conversations with any number of participants are built as a collection of simultaneous CP instances. In other words, the agent views a conversation as involving n participants while its interagent views such conversation as a collection of simultaneous dialogues represented as multiple CP instances.

Therefore, interagents constitute the sole and exclusive means through which CBR agents interact in our approach. Interagents are responsible for enforcing the protocols that guarantee that every CBR agent behaves according to the rules of an auction. As to agents playing the role of auctioneer, they must hold several conversations at the same time with the bidders in the auction that they govern. For this purpose, interagents must exploit their capability for supporting multiple conversations by building collections of simultaneous CP instances (one per bidder agent). Thus, for example, the auctioneer's interagent maintains an instance of the CP/DBP for each bidder agent in the auction.

3.1 Auctioning Diabetes Patient Cases

We are developing CoDiT, an agent-based system for therapy recommendation in diabetic patients lying in the framework of the SMASH project[6]. This agent-based system consists of a group of agents that perform case-based reasoning (CBR) and are able to communicate and cooperate for the purpose of recommending a therapy. Each agent has a case base with data of the patients of a specific M.D.; moreover, legal and deontological reasons prevents that patient data could be centralized since only the patient's doctor is entitled to have that data. Thus, this scenario fits the agent approach since resources are distributed but some doctors (or their agents) could also be interested in the case of a patient that is unknown to them but stored in some other doctors' case base. The doctors we are cooperating with have developed a common ontology to represent the patients data—thus we can focus on the cooperation and communication for the therapy task. Finally, remark that our diabetes therapy CBR agents are *peer agents* in the MAS, since each agent is capable of solving the whole task alone (recommending a therapy) using the resources available in its case base. However, it is intuitively appealing the scenario in which they are able to exchange patient data (maintaining anonymity for legal and deontological reasons) in order to improve their performance.

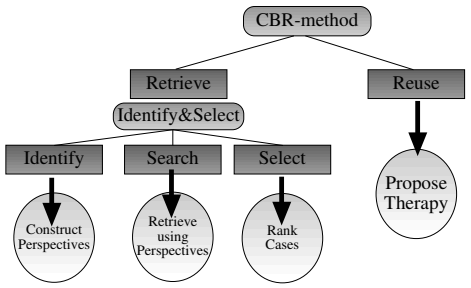


Figure 1: Task decomposition of the case-based reasoning method for diabetes therapy. The retrieve task will be augmented with communication capabilities for searching cases solved by other agents.

The basic CBR task decomposition is that shown in Fig. 1: retrieve and reuse. There is also an automatic retain task (not shown in the figure) that incorporates a solved problem into the agent's episodic memory. Summarily, the retrieve task has a decomposition method with three subtasks: identify, search and select. The identify task has a

[6]Systems of Multiagents for Medical Services in Hospitals, more information is available at http://www.iiia.csic.es/Projects/smash/

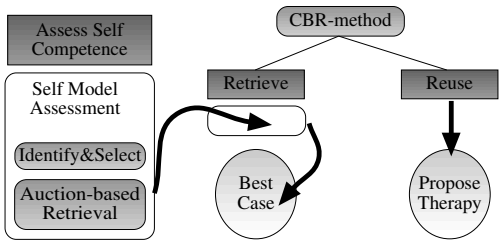


Figure 2: Task decomposition of the ABR for diabetes therapy. There is a metalevel task assess-self-model that decides whether to solve the retrieve task using the agent own method Identify&Select or the ABR method that will establish an auction with other agents.

method that constructs a perspective[7] on the patient; then task search retrieves from the case-base those cases that are subsumed by that perspective (i.e. those cases that satisfy the model built by the perspective). Next, the select task has a method that constructs a preference model of the retrieved cases from domain-specific knowledge. Finally, reuse is a task that takes the most preferred case and adapts its solution (therapy) to the current patient; the adaptation method uses domain-specific knowledge and if a most preferred case cannot be adapted then tries to adapt the next-preferred case.

The methods used in the retrieve task can incorporate communication and cooperation with other agents in order to find relevant cases known for other agents. Auction-based retrieval constitute a negotiation process to determine the peer agents with which is better to cooperate for solving a specific problem.

In CoDiT hypothetical prices are used in order to ensure a competitive market that motivates the participation of bidders. In other domain applications a real price can be used instead. Then an underlying payment mechanism among agents is needed.

4 Discussion

Some related approaches to case retrieval in a distributed setting are CBR-TEAM and NR. The CBR-TEAM [14] system allows cooperative retrieval and composition of a case whose subcases are distributed across several (specialist) agents in a MAS. This approach addresses the issues appearing on CBR in a distributed problem solving setting —centralized design and specialist agents— while we focus on a setting on peer agents with autonomy and decentralized design. A *Negotiation Retrieval* (NR) algorithm based on a distributed constraint optimization process is introduced to smooth inconsistencies by cooperative retrieval of cases while negotiating compromises to solve conflicts. Schematically, a complex query is presented to several agents. Then each agent is responsible for retrieving information related to a part of the query. Afterwards, agents negotiate to piece together an acceptable response. Again the difference of NR with our approach is the focus on centralized design and specialist agents.

The auction metaphor for multiagent case retrieval is a powerful one, and quite flexible since more than one auction protocol can be used, depending on the nature of the application envisioned. Let us suppose an application domain where cases can be freely sent over the network from one agent to another without violating any privacy or property conditions. In this situation, an alternative auction protocol can be used where the agent sends (a complete description of) the current problem and the bidders answer their case with best "degree of similarity" with respect to that problem. Then the auctioneer agent simply takes the higher bid (higher similarity). This is similar to *closed envelop one-round auction*, where a description of the proposed contract is sent and the bidders submit a description of an offer and a price, at the same time, and ignoring the offers of the other agents (the "closed envelop"). The difference from our proposed auction protocol is that here each bidder agent is free to use its own individual similarity criteria; the auctioneer has no control over them. Thus the auctioneer either a) has to trust on the competence levels of the other agents or b) the system is designed in such a way that all agents use the same similarity (centralized design hypothesis). On our Top-down Auction the auctioneer has control on the similarity criteria embodied on the ordered sequence of retrieval patterns.

Let us consider again the *closed envelop one-round auction*. This auction protocol can be modified to achieve that the auctioneer also has the "last say" capability. The only change we need is the following: the auctioneer does not accept automatically the "highest bid" asking first the case to the

[7] A perspective constructs from the patient model a specific model of the relevant aspects for the task at hand [1].

bidder in order to internally assess its quality according to the auctioneer's individual criteria of interest. If the case is valid the auctioneer pays the fee and otherwise opens a new auction where the failure case agent is not invited to join.

Which auction metaphor is better depends on the kind of application domain we are dealing with; however, the interesting issue is being able to accommodate and implement all or most of them. we think that is what the auction metaphor offers: the different auction protocols existing (and the variations that can be designed) can encompass a great variety of real life situations and can satisfy practical requirements of legal, economic, technical and nature.

Finally, another important issue is that each agent has limited information and individual interests. This is is the case in general and cannot be assumed otherwise because there is no way to prove or enforce that a MAS consists of uniformly biased agents[8]. For example, the auctioneer agent (be it the CBR agent recommending diabetes therapy or the consumer choosing a mortgage contract) has individual interests not shown in the "public declaration" proclaimed at the auction. Indeed, the auctioneer retains the ultimate decision power of committing to accept a specific bid without a public justification—if this were not so then an automatic decision procedure like "highest bid" would be all that is required in the auction-based retrieval.

Acknowledgments This work has been supported by the Spanish CICYT project SMASH, TIC96-1038-C04001; the COMRIS project, ESPRIT LTR 25500; Francisco J. Martin enjoys DGR-CIRIT doctoral scholarship FI-DT/96-8472.

References

[1] J. L. Arcos and R. López de Mántaras. Perspectives: a declarative bias mechanism for case retrieval. In D. Leake and E. Plaza, editors, *Case-Based Reasoning. Research and Development*, number 1266 in Lecture Notes in Artificial Intelligence, pages 279–290. Springer-Verlag, 1997.

[8] For particular systems, centralized design can be used to enforce that, e.g., all CBR agents have the same similarity criteria. however, this centralized solution is not applicable to all situations for a variety of legal, organizational and practical reasons.

[2] J. L. Arcos and E. Plaza. Inference and reflection in the object-centered representation language Noos. *Journal of Future Generation Computer Systems*, 12:173–188, 1996.

[3] M. Barbuceanu and M. S. Fox. Cool: A language for describing coordination in multi agent systems. In *Proceedings of the First International Conference on Multi-Agent Systems*, 1995.

[4] R. Davis and R. Smith. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence*, 20(1):63–109, Jan. 1983.

[5] P. Garcia, E. Giménez, L. Godo, and J. A. Rodríguez-Aguilar. Possibilistic-based design of bidding strategies in electronic auctions. In *The 13th biennial European Conference on Artificial Intelligence (ECAI-98)*, 1998.

[6] B. A. Huberman and S. Clearwater. A multi-agent system for controlling builging environments. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 171–176. AAAI Press, June 1995.

[7] F. J. Martin, E. Plaza, and J. L. Arcos. Knowledge and experience reuse through communication among competent (peer) agents. 1998. To appear in International Journal of Software Engineering and Knowledge Engineering.

[8] F. J. Martin, E. Plaza, and J. A. Rodriguez. An infrastructure for agent-based systems: an interagent approach. 1999. To appear in International Journal of Intelligent Systems.

[9] C. S. Noyda Matos and N. R. Jennings. Determining successful negotiation strategies: An evolutionary approach. In *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS-98)*, 1998.

[10] T. Ohko, K. Hiraki, and Y. Anzai. Addresse learning and message interception for communication. In G. Weiss, editor, *Distributed Artificial Intelligence Meets Machine Learning. Learning in Multi-agent Environments*, number 1221 in Lecture Notes in Artificial Intelligence. Springer-Verlag, 1997.

[11] H. V. D. Parunak. Visualizing agent conversations: Using enhanced dooley graph for agent

design and analysis. In *Proceedings of the Second International Conference on Multi-Agent Systems*, 1996.

[12] E. Plaza. Cases as terms: A feature term approach to the structured representation of cases. In M. Veloso and A. Aamodt, editors, *Case-Based Reasoning, ICCBR-95*, number 1010 in Lecture Notes in Artificial Intelligence, pages 265–276. Springer-Verlag, 1995.

[13] E. Plaza, J. L. Arcos, and F. Martín. Cooperative case-based reasoning. In G. Weiss, editor, *Distributed Artificial Intelligence Meets Machine Learning. Learning in Multi-agent Environments*, number 1221 in Lecture Notes in Artificial Intelligence. Springer-Verlag, 1997.

[14] M. V. N. Prasad, V. R. Lesser, and S. E. Lander. Retrieval and reasoning in distributed case bases. *Journal of Visual Communication and Image Representation, Special Issue on Digital Libraries*, 7:74–87, 1995. Also as UMASS CS Technical Report 95-27.

[15] J. A. Rodriguez-Aguilar, F. J. Martin, P. Noriega, P. Garcia, and C. Sierra. Competitive scenarios for heterogenous trading agents. In *Second International Conference on Autonomous Agents*, 1998.

[16] J. A. Rodríguez-Aguilar, F. J. Martín, P. Noriega, P. Garcia, and C. Sierra. Towards a test-bed for trading agents in electronic auction markets. *AI Communications*, 11(1):5–19, 1998.

[17] J. A. Rodríguez-Aguilar, P. Noriega, C. Sierra, and J. Padget. Fm96.5 a java-based electronic auction house. In *Second International Conference on The Practical Application of Intelligent Agents and Multi-Agent Technology(PAAM'97)*, 1997.

[18] R. G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, Dec. 1980.

[19] L. Steels. Cooperation between distributed agents through self-organization. In Y. Demazeau and J.-P. Müller, editors, *Decentralized A.I.*, pages 175–196. North-Holland, 1990.

[20] M. P. Wellman. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research*, (1):1–23, 1993.

[21] B. Wielinga, W. van de Velde, G. Schreiber, and H. Akkermans. Towards a unification of knowledge modelling approaches. In J. M. David, J. P. Krivine, and R. Simmons, editors, *Second generation Expert Systems*, pages 299–335. Springer Verlag, 1993.

[22] P. R. Wurman, , M. P. Wellman, and W. E. Walsh. The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agents. In *Second International Conference on Autonomous Agents (AGENTS'98)*, 1998.

[23] F. Ygge and H. Akkermans. Making a case for multi-agent systems. In *Proceedings of the Eighth Modelling Autonomous Agents in a Multi-Agent World (MAAMAW '97)*, pages 156–176, 1998.