Case Based Reasoning in Multiagent Systems

Chapter · January 2004

DOI: 10.1007/978-3-540-40003-5_7

Zhaohao Sun
The Papua New Guinea University of Technology

7 Case Based Reasoning in Multiagent Systems

This chapter is the second chapter in Part II of the book. It is also the basis for Chapter 9, as shown in the shaded area of Fig. 7.1. This chapter first examines the relationship between case-based reasoning (CBR) systems and multiagent systems (MASs), and proposes knowledge-based models of multiagent CBR systems from both logical and knowledge-based viewpoints. Then this chapter investigates the case base and case retrieval in a distributed setting and examines the integration of case-based reasoning capabilities in a BDI architecture. This chapter also discusses CBR for agent team cooperation. Finally this chapter proposes an agent architecture using CBR to model an agent negotiation strategy.

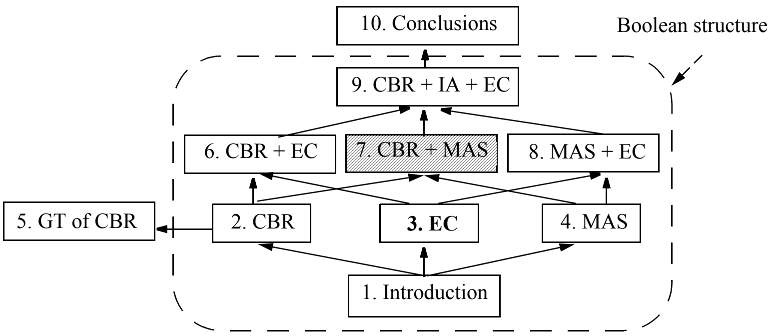

Fig. 7.1. Chapter 7 in the Boolean structure of this book

7.1 Introduction

Case-based reasoning (CBR) and multiagent systems (MASs) are two different paradigms in AI. CBR is a reasoning paradigm based on experience-based reasoning or similarity-based reasoning. MAS is a new paradigm to organise AI applications. However, integration of CBR and MASs has drawn increasing attention in the AI community [106][238][243], because CBR offers the multiagent systems paradigm the capability of autonomously learning from experience. Plaza and Ontañón [239] propose a framework for collaboration among agents that use CBR.

Prasad [243] discusses issues pertaining to cooperative retrieval and composition of a case in which subcases are distributed across different agents in a MAS. Giampapa and Sycara [106] discuss conversational case-based planning for agent team coordination, in which the acquisition and maintenance of the contextual information that determines the plan requirements is performed by a conversational case-based reasoner, NaCoDAE. NaCoDAE is also used to compositionally generate hierarchical task network plan objectives for the team agents with a MAS. Plaza et al. [238] investigate cooperation among agents that learn and solve problems using CBR. Further, Olivia et al. [226] describe a framework that integrates CBR capabilities in a BDI architecture.

One of the differences between these investigations is that some of them are in a heterogeneous environment, while the others are in the homogeneous environment. This chapter pursues this advance from a new perspective. That is, it first examines the relationship between CBR systems (CBRSs) and MASs, and proposes knowledge-based models of integrating CBRSs and MASs, which covers almost all attempts that apply CBR in MASs at a high level. Then it discusses how CBR has been applied in MASs at a concrete level. More specifically, this chapter will investigate the case base and case retrieval in a distributed setting, and examine the integration of case-based reasoning capabilities in a BDI architecture. This chapter will also discuss CBR for agent team cooperation. Finaly this chapter will propose an agent architecture using CBR to model an agent negotiation strategy

The rest of this chapter is organised as follows: Section 7.2 examines the relationship between CBR and MASs from both a logical viewpoint and a knowledge-based viewpoint and then proposes knowledge-based models for multiagent CBR systems integrating CBRSs and MASs. Section 7.3 investigates the case base and case retrieval in a distributed setting. Section 7.4 discusses CBR for agent team cooperation, which consists of a MAS and an agentified CBR system. Section 7.5 investigates the integration of CBR capabilities in a BDI architecture. Section 7.6 examines how CBR can improve cooperation among the agents within a MAS and how agents use CBR to learn from experience in a medical domain by discussing two cooperative modes for CBR agents within a MAS. Section 7.7 looks into the multiagent negotiation with CBR and the last section ends this chapter with some concluding remarks.

7.2 A Knowledge-based Model of Multiagent CBR Systems

Section 2.3 discussed the relations between expert systems (ESs) and case-based reasoning systems (CBRSs), while Section 4.6 investigated interrelationships of ESs and MASs. Therefore, this section integrates discussions in Section 2.3 and Section 4.6 and examines the relationship between CBRSs and MASs from both a logical viewpoint and a knowledge-based viewpoint and then proposes knowledge-based models of MASs which are based on both CBR systems and knowledge-based systems (see Fig. 7.2).

Section 2.3 concluded that CBRSs can be considered a further development of ESs. Further, both CBRSs and ESs rely on the explicit symbolic representation of experience-based knowledge to solve a new problem [197]. However, ESs use past

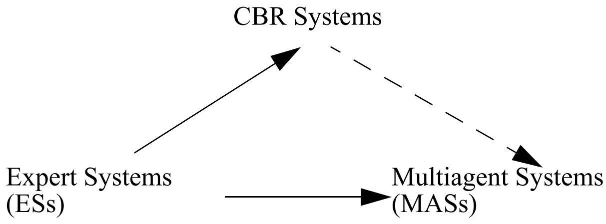

Fig. 7.2. ESs, CBR systems and MASs

experience stored in a knowledge base of generalized heuristics to assist in solving a new problem. They can store the generalized heuristics as rules of thumb or as logical inferences. CBRSs use an abstraction of specific problem-solving experiences to learn to solve a new problem. The representation of specific experiences usually includes the justification of the solution or the requirements of the problem as well as its solution. Moreover, compared to knowledge-based reasoning in ESs, CBRSs stress experience-based reasoning. The case base is an important component in CBRSs, while the knowledge base is one of the main components in ESs. Because similarity-based reasoning is an operational definition of experience-based reasoning, CBR can be considered as a kind of similarity-based reasoning from a logical viewpoint, while the CBRS is still a kind of knowledge-based system from a knowledge-based viewpoint. Therefore, the traditional CBRS can be briefly modelled as:

$$\text{CBRS} = \text{Case base} + \text{CBR engine} \tag{1}$$

where CBR engine denotes the inference engine in CBR system. The CBR engine performs deductive reasoning, in particular similarity-based reasoning, while inference engine in ESs performs traditional deductive reasoning. In this sense, the CBRS is a further development of an ES. This means that the CBRS is similar to the ES from a knowledge-based viewpoint; that is:

$$CBR = CB + CBRE \approx ES = KB + IE \tag{2}$$

where CB denotes the case base in the CBRS, while KB is the knowledge base in the ES. $CBRE$ denotes the CBR engine in the CBR system, while IE is the inference engine of the ES.

Section 4.6 investigated interrelationships of ESs and MASs and showed that high-level intelligence of a system requires a more complex system structure than low-level intelligence does in most cases. The intelligence level of the MAS can be improved through coordination, cooperation, communication, and negotiation among the agents within the MAS, although each of them may be less intelligent than an ES. It thus emphasized that simulation of human intelligence depends not only on the computerized knowledge and reasoning of human experts, to which ESs have paid much attention, but also on cooperation, coordination, communication, and negotiation among the components (agents) within an intelligent system, which MASs have emphasized. The above consideration was summarized in Section 4.6 as the following important relationship between ESs and MASs:

$$MAS = \sum_{1}^{n} A_i + C \approx \sum_{1}^{n} qES_i + C = \sum_{1}^{n} (KB_i + IE_i) + C \qquad (3)$$

where A_i is agent i within the MAS, $qES_i = KB_i + IE_i$ is the quasi-ES corresponding to agent i, $i = \{1, 2, \ldots, n\}$. C is the above-mentioned modules for coordination, cooperation, communication, and negotiation among the agents. $\approx$ stands for "is similar to". An concrete example for model (3) is the rule-based multiagent system MAGSY in [97]. Each agent in MAGSY has the problem solving capacity of an expert system and is defined by a triple (F, R, T), where

- F is a set of facts which represent the local knowledge of the agent
- R is a set of rules which define the strategies for the general behaviour of the agent
- T is a set of services which are provided by the agent.

Now taking into account (2), (3) becomes

$$MAS = \sum_{1}^{n} A_i + C \approx \sum_{1}^{n} qCBR_i + C = \sum_{1}^{n} (CB_i + CBRE_i) + C \qquad (4)$$

where A_i is still agent i within the MAS, $qCBR_i = CB_i + CBRE_i$ is the quasi-CBRs corresponding to agent i, $i = \{1, 2, \ldots, n\}$.

Therefore, a MAS can be viewed as a kind of CBRSs. Furthermore, it might be practical to simulate each agent within the MAS using CBR technology as much as possible, while one should make good use of MAS technology to deal with coordination, cooperation, communication, and negotiation among the agents in order to improve the intelligence of the MAS [94].

A concrete example of this model (4) is DistCBR and ColCBR in [238], in which all agents have CBR ability. Another example is a Web-based CBR agent for financial forecasting in [187]. Further, this model is also a more precise form for the multiagent CBR (MAC) system: $M = \{\{A_i, C_i\}\}_{i = 1, \ldots, n}$ proposed by Plaza and Ontañón [239], where M is composed of n agents, and each agent A_i has a case base C_i. Therefore, the above investigation can be considered a generalization of the models of Plaza et al. in [238], and Liu et al. in [187] as well as Plaza and Ontañón [239] for applying CBR in MASs.

The rest of this section will examine the (4) in some more detail:

1. If $CB_i \cap CB_j = \emptyset$ for $i, j = 1, \ldots, n, i \neq j$, then any different two agents A_i and A_j don't share common cases in their own case base. This means that the agents have different experience. This condition sometimes facilitates the corresponding experiments (see [239]) but it might affect the cooperation among agents
2. If $CB_i \cap CB_j = \emptyset$ for $i, j = 1, \ldots, n, i \neq j$ and $CBRE_1 = \ldots = CBRE_n = CBRE$, then the MAS degenerates to the Ensemble CBR system [239] in which CBR agents $A_1, \ldots, A_n$ work with the same CBR method but they have different experience (i.e. different case base $CB_1, \ldots, CB_n$)
3. If $CB_i \cap CB_j = \emptyset$ for $i, j = 1, \ldots, n, i \neq j$ and $CBRE_1 \neq \ldots \neq CBRE_n$, then the MAS is a model for the real world scenario in which CBR agents $A_1, \ldots, A_n$ work with different CBR methods, and they have also different experience (i.e. different case bases $CB_1, \ldots, CB_n$. It should be noted that the different CBR methods result from that CBR is a kind of similarity-based reasoning from a logical viewpoint. Different similarity metrics lead to different CBR methods or CBR engines. When $n = 2$, agent A_1 and A_2 work with different CBR methods and have different experience to negotiate over a series of negotiation issues (see Section 7.7)
4. If $CB_1 = CB_2 = \ldots = CB_n = CB$ and $CBRE_1 \neq \ldots \neq CBRE_n$, then the CBR agents within the MAS share a common case base, CB, but work with different CBR methods. This case usually happens in the real estate agency in which each CBR agent is a software counterpart of a human agent working in the real estate agency. They share the common resources of properties of houses in the real estate agency. However, they can use different CBR methods to negotiate with the customer over a certain property.

It should be noted that the above discussion is limited to some special cases in multigant CBR systems. In fact, the most general case is where some CBR agents within the MAS share a common case base, while other CBR agents have their own case bases. Some CBR agents like to work with the same CBR method, while other CBR agents work with different CBR methods.

Furthermore, it should be noted this model is homogeneous[1], because each of the agents within the MAS possesses the same ability; that is, CBR. This is not the real case in practice [222]. Therefore, it is necessary to propose the following model, which can be called heterogeneous,

$$MAS = \sum_{1}^{n} A_i + C \approx \sum_{1}^{m} ES_i + \sum_{m+1}^{n} CBR_i + C \tag{5}$$

$$= \sum_{1}^{m} (KB_i + IE_i) + \sum_{m+1}^{n} (CB_i + CBRE_i) + C$$

where, $1 < m < n$. A concrete example of model (5) is CoDiT, a MAS for case-based therapy recommendation in [203] (see Section 7.6). If $m = 1$, then another concrete example of model (5) is the RETSINA multiagent system in [106], in which a conversational case-based reasoner was agentified and inserted (see Section 7.4). This model will be used for integrating CBR and MAS in e-commerce in Chapter 9.

7.3 Distributed Case Base and Retrieval

In the model (4) of the previous section, the case bases $CB_1, \ldots, CB_n$ may not be situated at a single physical location and may be distributed across the agents A_i, $i = 1, \ldots, n$. For example, in the architecture of distributed CBR in [74] the case bases CB_i are distributed across client nodes and there is also a case base on a central server. How do distributed case bases arise in these MASs [243]? A system that performs rote learning by storing successful problem-solving episodes, where each agent A_i stores its own local case in its case base CB_i, could give rise to such a distributed case base (DCB). However, this may not be the only way, because each of the agents could acquire its own independent problem solving experiences by participating in different teams of agents. Another scenario that one could envisage now is the existence of case bases spreading across the Internet as Doyle et al. did in [74] and CMB, a multiagent CBR system for e-commerce [304]. In this situation, case bases for individual agents may be built independently, without complete knowledge of the kind of problem solving systems in which they are going to participate [242]. Central retrieval queries may not be satisfied by any one case base and may need a composite case derived from different case bases.

Reasoning about cases drawn from a case base that is a component of a DCB presents an agent with additional uncertainties versus single agent CBR systems [242]. As discussed previously, each agent has to rely on its possibly incomplete local view of problem solving to retrieve a local case that best contributes to the overall case. This may lead to the retrieval of subcases that cannot be effectively

1. This research stresses homogeneous agents have the same knowledge-based reasoning paradigms, while Plaza et al [238] believe that homogeneous agents have the same representation languages so that communication among agents does not require a translation phase.

put together or there may be requirements on the solution that cannot be ascertained until the subcases are aggregated. Thus, Prasad et al. [242] propose the negotiated case retrieval (NCR) strategy that needs the agents to augment their local views with constraining information from other agents to achieve the retrieval and assembly of a better overall case. This strategy involves that each agent asynchronously executes one of the set of possible operations: *initiate* a seed subcase, *extend* an existing partial case, *merge* existing partial cases or *inform* others about a new partial case, as shown in Fig. 7.3.

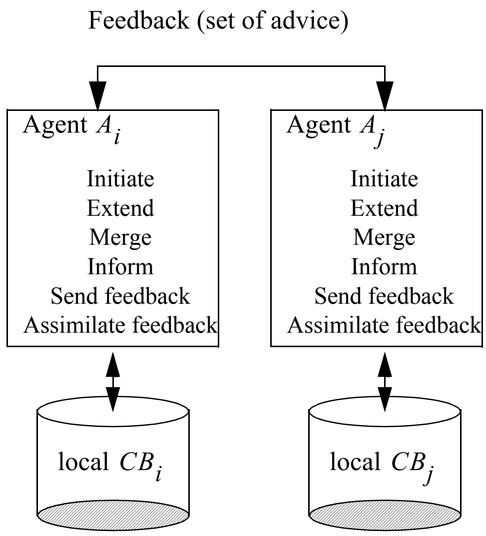

Fig. 7.3. Model of negotiated case retrieval after [242]

Initiating a seed subcase involves an agent retrieving a local subcase from its local case base, using the local problem solving state and the relevant portion of the user specification, and forming a seed subcase that can be extended by local cases from other agents to obtain a complete case [242].

An agent intending to *extend* a subcase from another agent obtains the subcase's relevant feature values that serve as an anchor for the local case retrieval, the result of which is integrated with the corresponding partial case [242].

Merge is similar to the extend operation. An agent intending to merge one of its chosen partial cases with another agent's partial case obtains the relevant feature values and performs the merge operation.

The *inform* operation involves an agent telling others about the existence of a newly formed partial case that results from the local execution of one of the three previous operators [242]. An *extend* or *merge* operation involves checking for any violations of local constraints by the set of feature values from the non-local partial case and the local case or partial case. Detection of such violations leads to an interaction process among the agents by which they negotiate on conflict resolution alternatives. The negotiation process involves an agent communicating feedback to other agents on the causes and possible resolutions for each of the constraint violations. The receiving agents assimilate this feedback, leading to an enhanced view of the global requirements for future operations. Any subsequent initiate, extend or merge is more likely to avoid the same conflicts.

7.4 CBR for Agent Team Coordination

As mentioned in Chapter 4, a MAS comprises a group of intelligent agents working together towards a set of common global goals or separate individual goals that may interact. In such a system, each of the agents may not be individually capable of achieving the global goals and/or their goals may have interactions- leading to a need for cooperation among the agents [243]. This section examines the agentification of a CBR system, NaCoDAE, into a multiagent system (MAS), RETSINA (reusable environment for task-structured intelligent network agents), in which the agents may use capability-based or team-oriented agent coordination strategies, for agent team coordination.

RETSINA is a collection of heterogeneous software entities that collaborate with each other to either provide a result or service to other software entities or to an end user [106]. Based on functional viewpoint, RETSINA agents are classified into four types: interface agents, task agents, middle agents, and information agents. RETSINA agents typically use the capability-based coordination technique to task each other, which means that one agent will dynamically discover and interact with other agents based on their capability descriptions. RETSINA agents also support other forms of coordination techniques, such as the team-oriented coordination.

NaCoDAE is a conversational CBR system that helps a user decide a course of action by engaging him in a dialogue in which he must describe the problem or situation [106]. A conversational session begins with the user providing an initial partial description of the problem that he tries to solve. NaCoDAE responds by recommending the ranked solutions from the case base, whose problem descriptions best match the user's problem descriptions, and the ranked questions, which are the unanswered questions in these cases, to the user (interface). After the user obtains these recommendations, he will either refine their problem description by answering selected questions, or accept a solution from the recommended solutions. Therefore, from a viewpoint of CBR, NaCoDAE has performed case retrieval, case recommendation, and problem adaptation that is a part of case adaptation.

NaCoDAE has three features that made it suitable for team co-ordination and interaction with RETSINA agents [106]. First, NaCoDAE can work with partial descriptions of the problem and use them for initiating a dialogue. This could allow one to encode a general strategy of “always knowing the strategy for how to get more information, if nothing else is known”. Second, NaCoDAE can continually revise its list of most likely candidate cases, as data is provided to the system by either an agent or the user. This feature leads itself to a form of coherent, compositional and incremental construction of knowledge structures, such as hierarchical task network (HTN) plan objectives and representations of situational or contextual knowledge. This knowledge can be accessed even if time and the lack of specific information do not allow for a description to be completely specified. Third, the cases can be modified to store any type of textual data, including agent capabilities and queries.

After agentification, the NaCoDAE becomes a RETSINA task agent [106] who is situated in the RETSINA community, where there are also Briefing Agents, Matchmakers, MissionAgents, VoiceAgents etc. They work together to perform a certain mission. The agent communication that involves BriefingAgent and NaCoDAE are run in the following way [106]: As the Company Commander speaks, his speech is translated into text by the VoiceAgent. The BriefingAgent receives those textual translations and attempts to match the text of the Commander's speech with the textual answers to questions that were posed by NaCoDAE. If there is a match, then the BriefingAgent will send that answer to NaCoDAE. If NaCoDAE can use that answer to complete a case, then it will return a case to the BriefingAgent; otherwise return a regenerated ranked list of questions and their associated answers. If NaCoDAE's questions contain agent queries, the BriefingAgent will directly query the provider agent if it is known, or first ask either or both of the Matchmakers for the identity of a provider agent, and then contact it. Upon request of the MissionAgents, or upon the completion of a case by NaCoDAE, the BriefingAgent will assemble a shared plan from the case actions and send it to the MissionAgents. During the execution of the scenario, the MissionAgents may also provide the BriefingAgent with updates to their capabilities, which the BriefingAgent can forward to NaCoDAE.

This section examined the CBR system for the team coordination of independent, intelligent software agents. According to Giampapa and Sycara [106], NaCoDAE has demonstrated that its conversational nature is well-suited for agent information gathering domains.

7.5 Case-based BDI Agents

Integrating CBR capabilities in a BDI architecture is another attempt to integrate CBR and MASs. This section will examine a framework that integrates CBR capabilities in a BDI architecture as well as its application to the design of Web information retrieval proposed by Olivia et al. [226].

BDI structure mainly consists of five factors: beliefs, desires, intentions, goals, and plans, which constitute the mental state of a BDI agent [35] (p 47), as shown in Fig. 7.4:

- Beliefs contain the fundamental views of an agent with regard to its environment. An agent uses them to express its expectations of the possible future states
- Desires are derived directly from the beliefs. They contains the agent's judgements of future situations
- Intentions are a subset of the goals. If an agent decides to follow a specific goal, this goal becomes an intention
- Goals represent that subset of the agent's desires on whose fulfilment it could act. In contrast to its desires, an agent's goals must be realistic and must not conflict with each other
- Plans combine the agent's intentions into consistent units.

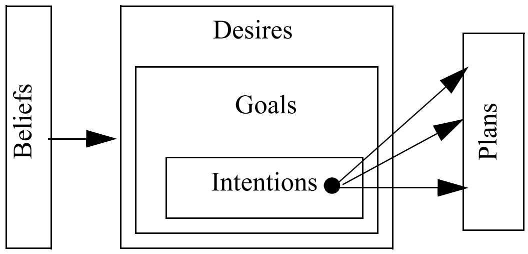

Fig. 7.4. BDI structure based on Brenner [35]

BDI agents have been widely used in relatively complex and dynamically changing environments [35]. Olivia et al. [226] proposes a CBR-BDI agent architecture for information retrieval (IR) on the WWW in order to improve the performance of currently deployed Web IR systems in terms of search efficiency and resource discovery in well-demarcated domains. Web CBR-BDI agents are designed to locate and extract information from homepages of academic staff members with particular research interests. The CBR-BDI architecture has the following main components: the case memory, the domain-specific knowledge base, and the CBR-BDI interpreter, as shown in Fig. 7.5 In what follows, the first two mentioned components will be examined in some detail.

- The domain-specific knowledge base is implemented in a form of concept hierarchy, which is collection of keywords representing broad areas of expertise (concepts). A collection of keywords (sub-concepts) representing specific subarea of expertise is also attached to these concepts. Concepts are mapped to specific university academic entities on a well-demarcated application domain such as Australian universities. The concept hierarchy plays an important role in focusing on the search process in the start-up of the Web CBR-BDI agent where no previous cases are stored, and when the similarity-based mechanism does not target any particular case in the case memory. Furthermore, it helps the system to identify related research interest if it fails to retrieve an exact match information. For example, if no exact match for research interest in mobile agents is found, the system is able to retrieve academic homepages with relevant research interest such as intelligent agents and autonomous agents

- Case memory. Cases stored in the case memory are constructed in terms of belief, desire, intention, outcome status, and outcome URL. More specifically, belief is the university domain to be searched. Desire is a sub-concept/concept that represents the specific/similar research interest being searched. Intention is the focused search to concept-related academic entities. Outcome status is either a successful or unsuccessful case. Outcome URL is the URL staff link directory academic entity.

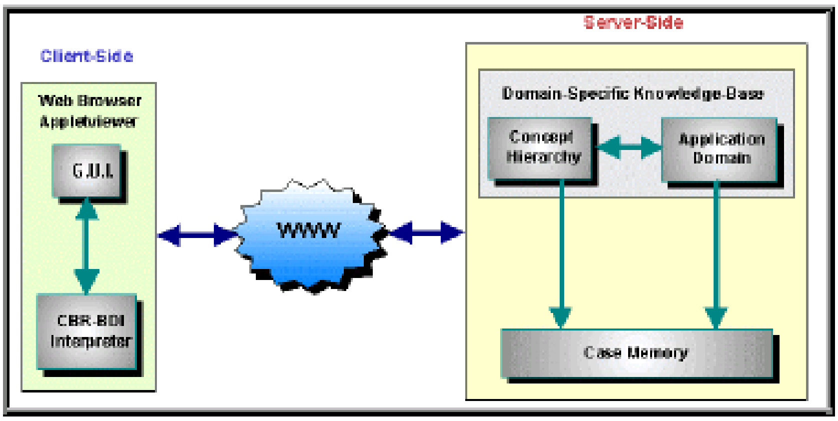

Fig. 7.5. A CBR-BDI architecture after [226]

The overall process is run as follows: The end-user is presented with the GUI where he can specify sub-concepts or research interests associated with a given concept/domain of knowledge, together with the universities of interest. Web CBR-BDI agents are triggered by pressing the start button. The first objective of the agent is to perform a standard CBR analysis of the input problem description. The input problem description is constructed by the combination of an end-user's selected university domain (belief), and subconcept/specific research interest (desire). The similarity-based mechanism serves to find the most similar cases with the input problem description.

Given that similar cases are sorted by outcome status (found/not found), the Web CBR-BDI agent first scans the most promising URLs (outcome status = found), and leaves for the last stages of the search the less promising ones (negative cases).

In the case where the similarity-based mechanism retrieves similar cases, the case memory may lead directly to a promising URL from where to initiate either a depth-first or breadth-first search, instead of traversing exhaustively the sub-webs of a particular university.

The results obtained from the CBR analysis drive the Web traversal of the agent to retrieve the desired information.

7.6 Cooperative CBR Agents in MAS

As mentioned in previous Chapter 4 and Section 7.4, cooperation is an important characteristic in MASs. An agent with "perfect" knowledge and "complete" capabilities for a given task has no need to require the cooperation of other agents [238]. However, a regular agent is less intelligent than an expert as discussed in Chapter 4, he can't have "perfect" knowledge and "complete" capabilities for a given task. Even an expert in a society can't say that he has "perfect" knowledge and "complete" capabilities for a given task in his professional field. Therefore, it is necessary for an agent to cooperate with other agents within the MAS to perform a given task. This section examines how CBR can improve cooperation among the agents within a MAS and how agents use CBR to learn from experience in a medical domain, which used to be an important application field of expert systems such MYCIN [299]. The real world scenario is CoDiT.

CoDiT is a MAS, wherein agents use CBR to recommend therapy for diabetic patients [203]. CoDiT consists of a few agents that perform CBR and are able to communicate and cooperate for recommendation of a therapy[1]. Each agent, as a software counterpart of a human doctor, has a case base with data of the patients of a specific M.D.; moreover, legal and deontological reasons prevent that patient data could be centralised since only the patient's doctor is entitled to have that data. Thus, this scenario fits the MAS 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 doctor's case base. Further, the diabetes therapy CBR agents in the MAS are *peer* agents, 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 obvious that in such a scenario the agents should exchange patient data (maintaining anonymity for legal and deontological reasons) in order to improve their performance.

The main CBR task involved in CoDiT is *retrieve* and *reuse* [203], as shown in Fig. 7.6. There is also an automatic **retain** task (not shown in the figure) that incorporates a solved problem into the agent's episodic memory. Generally speaking, the main retrieve task can be decomposed into three subtasks: identify, search, and select (also see [1]). The **identify** task has a method that constructs a perspective on the patient; then task **search** retrieves from the case base 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

---

1. For more information see http://www.iiia.csic.es/Projects/smach

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. This means that case adaptation has been used here as a part of case reuse. In what follows, the rest of this section examines how the methods used in the retrieve task can incorporate communication and cooperation with other agents in order to find out relevant cases for other agents.

A cooperation mode establishes how two agents must behave to accomplish a particular task [203]. However, when an agent can opt for more than acquaintance to cooperate in solving a specific (sub)task, then different co-operation strategies can be established for each cooperation mode depending on different criteria followed by the agent to solve such (sub)task. For instance, depending on how the set of helper agents chosen to cooperate is constructed and how this set is sorted to be traversed in search of a competent agent. In this way, a cooperation mode can determine how two agents cooperate whereas a cooperation strategy settles how more than two agents do.

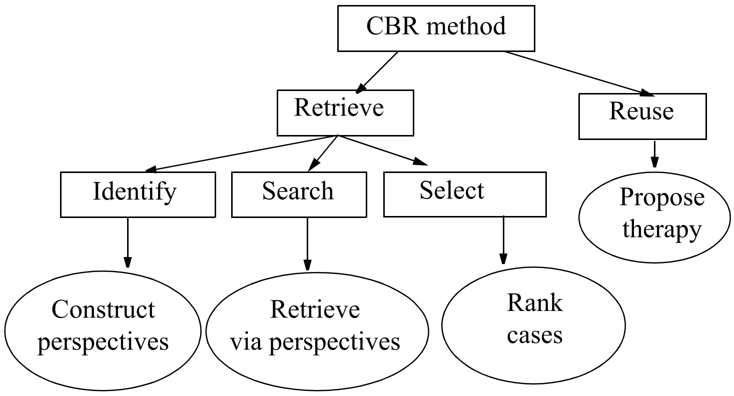

Fig. 7.6. Task decomposition of CBR for diabetes therapy after [203]

Therefore, the term cooperative CBR groups together the set of cooperation modes and cooperation strategies that can be deployed by some CBR agents wherein each CBR agent has its own case base [203].

Cooperation among CBR agents can be thought as an extension of agents' set of precedents; that is, an expansion of the individual memory of a CBR agent to the memories of some CBR agents [203]. For instance, in CoDiT the retrieve task incorporates cooperation with other agents in order to find relevant cases known for other agents- i.e. to find the patient record most relevant to the current problem.

Two cooperation modes between CBR agents were proposed in [203]: Distributed Case-Based Reasoning (DistCBR) and Collective Case-based Reasoning (ColCBR). The DistCBR cooperation mode is a class of cooperation protocols where a CBR agent A_{orig} is able to ask one or several other CBR agents $\{A_1, \ldots, A_n\}$ to solve a problem on its behalf, and the ColCBR cooperation mode is a class of cooperation protocols where a CBR agent A_i is able to send a specific CBR method to one or several CBR agents $\{A_1, \ldots, A_n\}$ that are capable of using that method with their case base to solve the task at hand [238]. Therefore, the DistCBR cooperation mode enables an agent to share experiential knowledge acquired by an acquaintance by means of particular problem solving methods, while the ColCBR cooperation mode allows a couple of CBR agents to share experiential knowledge. Both DistCBR and ColCBR are based on solving the retrieve task reusing the experiential knowledge (in form of cases) of other CBR agents:

- DistCBR. An agent (the originator) delegates the retrieve task to another agent (the helper) indicating the helper's CBR method to solve such task. In this sense, the CBR process is distributed since every agent works using its own method of solving problems [238]
- ColCBR. An agent (the originator) forwards the retrieve task and the PSM (problem solving method) of that task to an acquaintance (the helper). That is to say, the originator, in addition to the task, also conveys the PSM to solve that task. In this sense, the originator is using the memory of the other agents as an extension of its own- as a collective memory- by means of being able to impose to other agents the use of the CBR method of the originator [238].

In both cooperation modes helper's experiential knowledge is shared and then reused by the originator [203]. However, while the DistCBR cooperation mode also allows helper's problem solving knowledge to be shared and reused by the originator, using the ColCBR cooperation mode the PSM sent by the originator is shared by the helper to retrieve the most relevant case(s) that will be later reused by the originator. From an authority point of view, it can be said that using DistCBR the originator delegates authority to the helper to solve the task in hand. On the contrary, using ColCBR the originator maintains the authority, since it has fully control over the PSM applied, merely using the experiential knowledge of the helper.

The following actions are performed by two CBR agents whilst cooperating using the DistCBR cooperation mode [203]:

1. The originator asks the helper to solve (delegates) the retrieved task indicating which helper's problem solving method must be applied to solve such task
2. On receipt of the task, the helper retrieves the most relevant precedent(s) using its corresponding retrieval method (as indicated by the originator)
3. Thereafter, the helper refers the available precedent(s) back to the originator which will have been inferred using its own (helper's) PSM.

The ColCBR cooperation mode implies the following actions to be carried out between two CBR agents [203]:

1. The originator sends the retrieve task to be solved and a originator's retrieval method to be applied to solve such task together to the helper

2. On receipt of the task and the PSM, the helper retrieves the most relevant precedent(s) using the PSM received
3. Thereafter, the helper refers the available precedent(s) back to the originator which will have been inferred using the originator's PSM method.

This section discussed two different cooperative modes for CBR agents within a MAS. The above discussion allows us to exemplify the sharing and reuse of problem solving knowledge and experiential knowledge (in the form of cases in CBR) among agents within a MAS.

It should be noted that from a viewpoint of pure CBR, CoDiT is a case-based recommendation system for a medical domain, in particular for diabetic patients. The basic difference of CoDiT from other case-based recommendation systems mentioned in Chapter 6 is that CoDiT is placed in multiagent settings. Because of communication and cooperation, case-based recommendation systems become more complex in multiagent settings.

7.7 Applying CBR to Multiagent Negotiation

As mentioned in Chapter 6, an intelligent agent should be used to negotiate with the customers (or customer agents) for their demands and to assist them during the search for an appropriate product. This section will examine how to use CBR to automating negotiation in a multiagent setting.

7.7.1 Introduction

Negotiation in MASs is one of the main research lines in MASs and has been studied from many different points of view such as game theory, artificial intelligence, and CBR in Chapter 6. In the area of CBR, Sycara presents a model of negotiation that combines CBR and optimization of multi-attribute utilities of intelligent agents. She provides a model of goal conflict resolution through negotiation implemented in the PERSUADER system that resolves labour disputes. Matoes [205] employs CBR to determine in each step of the negotiation the best performance of the agent by selecting the weighted proposal combinations and the parameters associated with a set of tactics. Recent growing interest in intelligent agents in e-commerce has given more importance to the problem of automated negotiation. Intelligent agents negotiate to coordinate their activities and come to mutual agreement, in particular in the e-bargaining process (see Chapter 8). In many cases, automated negotiation requires different behaviours of intelligent agents for different negotiation situations [205]. The rest of this section will first look at the negotiation in a real estate agency and negotiation strategies. Then it will present an agent architecture using CBR to model an agent negotiation strategy.

7.7.2 Negotiation in a Real Estate Agency

In a real estate agency there is a set of properties that need to be sold. In this domain there are two main players: seller agents and buyer agents. The seller agent acts on behalf of the interests of the real estate agency, while the buyer agent represents the interests of a customer. The seller agent needs to sell a house with maximal profit for the agency, while the buyer agent wants to buy a house for his buyer with specific features and minimal price. This is an obvious conflict of interest that is usually resolved by a negotiation.

The seller agent has complete information of all the properties about houses on sale at the real estate agency. However, in some cases the buyer agent does not have a clear opinion on his preference on the negotiation issues. During the negotiation, the seller agent usually includes new negotiation issues to enrich the description of a house. Then the buyer uses this new information to compare and discriminate better among the different offers made by the seller agent. Thus, the buyer agent tries to obtain a complete description of the properties, negotiating over the set of negotiation issues mentioned before. Usually the agents try to adjust either the issues related to the description of the house and later the price issues. They negotiate until they obtain an agreement, in this case a property that satisfied both sides, if any exists, or one of them withdraws.

7.7.3 Negotiation Strategies

The negotiation strategies are defined based upon the knowledge, past experience, and information available to the negotiation agents [375]. The aim of a negotiation strategy is to determine the best courses of action to reach an agreement [205]. When agent a receives an offer from agent b, it becomes the last element in the current negotiation thread between the agents. If the offer is unsatisfactory to a, the agent a generates a counter-offer. In generating its counter-offer, a may use the information of mental state and different weighted combinations of tactics for each of the negotiation issues. The negotiation issues in the real estate agency mainly include the features of a house, for instance, surface, district, number of rooms, floor number, garage, price, brightness, number of bathrooms, elevator, and address [205].

Most systems use a number of predefined negotiation strategies to generate counter-offers. For example, negotiation in Market Maker [215] allows the agents to use three predefined negotiation strategies: anxious, cool-headed, and frugal corresponding to linear, quadratic, and exponential functions in the generation of offers/count-offers. The users need to decide which strategy the agents will follow during negotiation. However, negotiation strategies can also be acquired from previous negotiation cases or experiences based on CBR in the CBN (case-based negotiation) framework [375]. The CBN agents revise and adapt negotiation strategies in each decision-making episode of the negotiation process [375].

7.7.4 Case-based Agent Architecture

As mentioned in previous chapters, CBR has received a lot of attention over the last few years, and has been employed with good results in many areas [205][336] including negotiation in e-commerce. The case-based negotiation agent uses CBR technology to perform negotiation on behalf of either seller or buyer or broker; that is, he will assess at the similarity of the current negotiation to previous negotiation cases kept in the negotiation case base. The successful negotiation cases that the case-based negotiation agent performed are kept in the negotiation case base for *reuse* in later negotiation case retrieval. The case-based negotiation agent can use the fuzzy rule-based adaptation to adapt the most similar negotiation case to the current negotiation situation. The architecture of the case-based negotiation agent is shown in Fig. 7.7. Some components will be discussed in more detail.

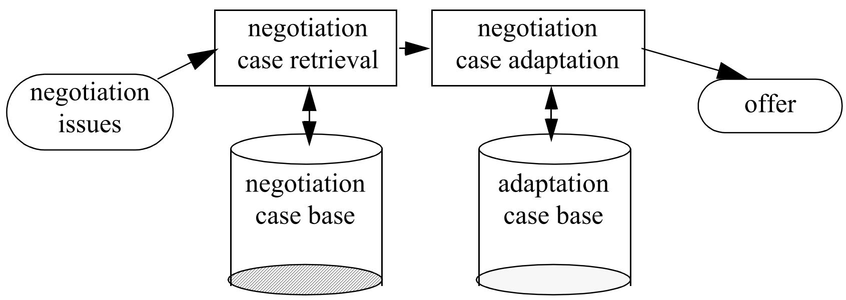

Fig. 7.7. An architecture of a case-based negotiation agent

- Negotiation issues are similar to the current problem in the traditional CBR model. It is also the requirements of a customer. For example, during the negotiation in a real estate agency [205], the seller agent and buyer agent will negotiate over the following negotiation issues for a house: price, number of rooms, garage, floor number, number of bathrooms, address, surface, district, furnished or unfurnished
- A negotiation case in the negotiation case base can be considered as a prior successful negotiation process, which mainly includes the sequence of offer and counter-offer and the eventual successful offer based on the initial negotiation issues. It also provides detailed negotiation context and decisions made in previous negotiations

- The negotiation case retrieval is executed concurrently with the other activities of a case-base negotiation agent. When an agent sends an offer, it immediately begins to retrieve those negotiation cases that are most similar to the current negotiation cases from its negotiation case base. When it receives a counter-offer corresponding to its offer it is incorporated into the negotiation thread and used to finally select the most similar negotiation case from those that were obtained in the meantime
- Once a negotiation case is selected as the most relevant negotiation case to the current negotiation issues, the case-based negotiation agent might revise or adapt the negotiation case in order to meet the changing count-offer from his counterpart. The negotiation case adaptation can depend on a set of fuzzy adaptation rules, which represent conditions of the environment in which the negotiation acts and determines variations in the value of the parameters of the negotiation issues and negotiation tactics (also see [205]). In general, these fuzzy rules follow the following classical form:

$$Rule_i: \text{IF } x_1 \text{ is } A_{i1} \text{ and } \dots \text{ and } x_n \text{ is } A_{in} \text{ Then } y \text{ is } B_i \tag{6}$$

where $x_1, \dots, x_n$ and y are the feature variables, $A_{i1}, \dots, A_{in}$, and B_i are linguistic labels of the variables $x_1, \dots, x_n$, y which are in the universe of discourse $U_1, \dots, U_n$, V of the variables. An example of linguistic labels might be: {excellent, good, not satisfactory, bad}. These linguistic labels are characterised by their membership functions $\mu_{A_{ij}}: U_j \to [0, 1]$, $j = 1, \dots, n$; $B_i: V \to [0, 1]$.

7.8 Concluding Remarks

This chapter examined the relationship between CBR and MASs from both a logical viewpoint and a knowledge-based viewpoint and then proposed knowledge-based models of a multiagent CBR system integrating CBR systems and knowledge-based systems, which basically cover almost all attempts that have applied CBR in MASs in a homogeneous or heterogeneous setting. The key idea behind these models are that CBR systems can be considered as a further development of expert systems (ESs), and the integration of CBR systems and MASs should take into account cooperation, coordination, communication, and negotiation, in order to model the social function of individual intelligence. Then this chapter investigated the case base and case retrieval in a distributed setting and examined the integration of case-based reasoning capabilities in a BDI architecture.

Cooperation is an important characteristic in MASs. This chapter discussed CBR for agent team cooperation, which consists of a MAS and an agentified CBR system, and examined how CBR can improve cooperation among the agents within a MAS and how agents use CBR to learn from experience in a medical domain by discussing two cooperative modes for CBR agents within a MAS.

Negotiation is another important characteristic in MASs. This chapter looked into CBR-based negotiation in a real estate agency and negotiation strategies. Then it proposed an agent architecture using CBR to model an agent negotiation strategy.

It should be noted that research and development of multiagent CBR systems is still in its infancy, although some advances in this field has been reported or appeared in the past few years. Further, the studies are basically in a homogeneous multiagent setting. Therefore, there are a lot of issues in the future study of multiagent CBR systems. For example, the proposed models ((4) and (5)) require further investigation at a more detailed level. Negotiation is a general concept in a multiagent setting. In fact, its special forms are auction, brokering, and mediation, which are all important for commerce and business. Therefore, how to apply CBR in auction, brokering, and mediation in a multiagent e-commerce setting is a big issue for intelligent e-commerce, which will be examined in Chapter 9.

Finally, as we argued in Chapter 4, coordination, cooperation and communication are important components in multiagent systems. All these three are social behaviours in human society. Experience plays an important role in these social behaviours, and CBR is a kind of experience-based reasoning. Therefore it is still significant to examine applications of CBR in coordination, cooperation and communication from a multiagent system viewpoint, although we have reviewed the cooperative CBR agents in [6].