A Support System for Home Care Service Based on Multi-agent System

Goichi ITABASHI
SCIENTIA Corporation,
SENDAI, 981-3204 JAPAN
Email: itabashi@scientia.co.jp

Mikiko CHIBA, Kaoru TAKAHASHI and Yasushi KATO
Sendai National College of Technology,
SENDAI, 989-3128 JAPAN
a0535@bamboo.sendai-ct.ac.jp,
kaoru@cc.sendai-ct.ac.jp,
kato@info.sendai-ct.ac.jp

***Abstract*— According to the increase of high quality welfare services in Japan, welfare services need engineering supports. In this paper, we develop a support system for a home care service, using a multi-agent system. The decision of a care schedule is performed autonomously by the negotiations among agents so that it is sufficient for clients and helpers to confirm schedules settled by the agents. They can confirm the schedules using PDAs, which can be easily handled even by elders. Consequently, it is expected that our support system reduces the total cost of a home care service.**

***Keywords*—home health care, scheduling, multi-agent system, PDA.**

I. Introduction

As the needs of high quality welfare services are increasing in Japan, welfare services need engineering supports. We develop a support system for a home care service, by using a multi-agent technique[1, etc.]. In our support system, the decision of a care schedule is performed autonomously by the negotiations among agents. It is sufficient for clients and helpers to confirm schedules settled by the agents. A personal digital assistant (PDA) is used as an interface between an agent and a client/helper. A PDA enables helpers and elders to confirm schedules easily, because a PDA can be handled by tapping its screen with a stylus or even a fingertip. Consequently it is expected that our support system reduces the total cost of the service and can improve the efficiency and quality of the service.

This paper is structured as follows. We outline a home care service in Section II. The design and implementation of our support system are discussed in Section III and Section IV, respectively. We have a plan to intoroduce an ontology into the support system. This plan is explained in Section V. Finally, we conclude this paper in Section VI.

II. Home Care Service

In a *home care service*, a *helper* visits a *client*'s house and provides various services such as *nursing care* and *living support*. A *care manager* investigates the mental and physical conditions of the client and makes a *long-term care plan*. Then, he/she requests care services to a *home care support center*. The home care support center assigns care schedules to an appropriate helper. The client must be certified as the person who needs a nursing care or a living support on *the long-term care insurance system* in Japan. A client who needs a nursing care gets nursing care services such as assistance for excretion, bathing, meal and walking. A client who is physically handicapped in daily life gets living support services such as cleaning, washing and cooking. Qualifications are required for the services. Helpers must have at least the third grade of the helper qualification to give living support services, and the second or first grade of the helper qualification is required for nursing care services.

In this paper, we develop a support system for the decision of care schedules mentioned above.

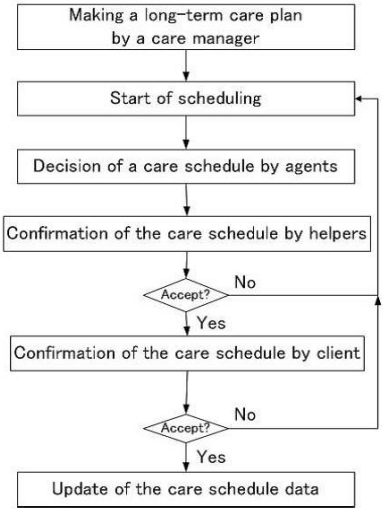

Fig. 1. Scheduling flow.

III. System Design

A. System Requirement

In this section, we design a support system for the scheduling of care services and the confirmation of a scheduling result by clients and helpers. The scheduling and the confirmation are done according to the flow shown in Fig. 1.

B. System Design

When a care manager makes a long-term care plan, he/she requests for care services to a home care support center. Care

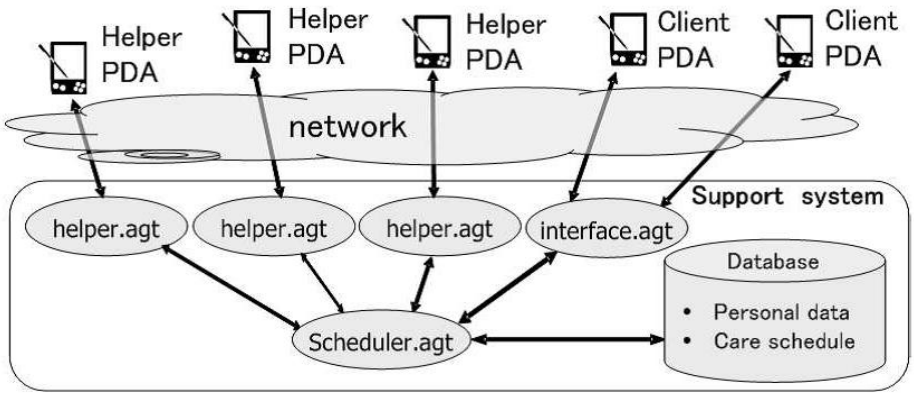

Fig. 2. System configuration.

schedules are made in the center and input to the system. Then, agents in the system start to negotiate the care schedules. The result of the negotiation is proposed to helpers. If the helpers accept the result, it is also proposed to the client. If the client accepts the result, the schedule of care services is fixed and the schedules of a client and a helper are updated. If the helpers or the client reject the result, the schedules are reconstructed. The follwing functions are required considering the flow in Fig. 1.

- The scheduling function
- The interface function between an agent and a client/helper
- The schedule management function

The interface between the agent and the client/helper is implemented on a PDA. Use of the PDA is easy for elders or people who are not familiar with the computer, because it can be handled by tapping its screen with a stylus or even a fingertip. Clients and helpers can communicate with the agents via their PDAs and can accept or reject the proposed schedules.

We design a support system based on the above system requirement. Fig. 2 shows the configuration of the system.

Clients and helpers have PDAs and communicate with the support system, which has a database and three types of agents called *interface agent*, *scheduler agent* and *helper agent*. The database manages the personal information and the long-term care plans for clients. The agents communicate with each other and propose care schedules to the helpers and the clients. Our support system is implemented as a multi-agent system. The agents communicate with each other based on the FIPA contract net interaction protocol[2]. The roles of the agents are presented in the following.

- Interface agent: this agent behaves as an interface between a client/the system administrator and the system. It communicates with the client via a PDA. It also communicates with the scheduler agent when the system administrator requests to plan a schedule.
- Scheduler agent: this agent manages a scheduling process and corresponds to the Initiator in the FIPA contract net interaction protocol.
- Helper agent: this agent is assigned to each helper and holds a schedule and personal information of the helper. The scheduling of care services is performed by helper agents and the scheduler agent. A helper agent communicates with a helper via a PDA. It corresponds to the Participant in the FIPA contract net interaction protocol.

The scheduling of care services is performed by the negotiation among agents. We show the scheduling process in the following.

1) When the interface agent receives a scheduling request, it sends a *REQUEST* message to the scheduler agent in order to tell the start of scheduling.
2) If the requested schedule contains a helper who the client chooses, the scheduler agent sends a *CFP* (call for proposal) message to the helper agent. If not, the scheduler agent sends a *CFP* message to helper agents whose owners have charge of the client.
3) When the helper agent receives the *CFP* message, it checks helper's schedule. If the date and time described in the *CFP* message are available on the schedule, the helper agent returns a *PROPOSE* message to the scheduler agent. If not, it returns a *REFUSE* message.
4) The scheduler agent sends the *ACCEPT_PROPOSAL* message to the helper agent, if the scheduler agent receives the *PROPOSE* message from the helper agent whose owner is chosen by the client. If not, the scheduler agent refers to the database to get total time of working. Then, the scheduler agent chooses one helper agent based on the referred data and sends an *ACCEPT_PROPOSAL* message to it.
5) When the helper agent receives the *ACCEPT_PROPOSAL* message, it requests the helper to confirm the care schedule. The detailed schedule is sent to the helper's PDA and displayed. The helper accepts or rejects the displayed schedule. If the schedule is accepted, the helper agent receives the result of the confirmation from the PDA, and sends an *INFORM* message to the scheduler agent. If not, the helper agent sends a *CANCEL* message to the scheduler agent.
6) When the scheduler agent receives the *INFORM* message, it sends an *INFORM* message to the interface

agent. Then, the interface agent requests the client to confirm the care schedule with the client's PDA. If the client accepts it, the interface agent sends an *AGREE* message to the scheduler agent. If not, it sends a *CANCEL* message to the scheduler agent.

7) When the scheduler agent receives the *AGREE* message, it updates the schedule data in the database. To complete the scheduling, the scheduler agent sends a PROPAGATE message to the helper agents that sent the *PROPOSE* messages.
8) When the schedule is not fixed due to the *REJECT* or *CANCEL* message, the scheduler agent tries to find a sub-helper set, which contains helpers who do not have charge of the client listed up on the requested schedule. The scheduler agent chooses several helpers from the sub-helper set, considering their qualification and total time of working. Then, the scheduler agent starts scheduling with the helper agents. If the negotiation with all helper agents is failed, the system terminates scheduling and starts again from the beginning using other conditions which are input by the system administrator.

Clients and helpers confirm schedules on their PDAs. In our support system, the interface/helper agent sends a proposed schedule to a client/helper, then receives the confirmation result. The followings are required as PDA functions.

- Communication with agents: PDA can communicate with agents in the support system via wireless LAN.
- The receipt of schedule data: PDA periodically communicates with the support system and receives schedule data. If PDA receives schedule data, it notifies the receipt with an alarm.
- Acceptance or rejection of a schedule: clients and helpers can accept or reject a proposed schedule in this function.
- Update of a schedule: schedules in PDA are updated with accepted schedule.

IV. Implementation

We have implemented a prototype of the support system based on the above mentioned system design. In our support system, multi-agent system has been implemented using JADE (Java Agent DEvelopment Framework)[3]. The PDA functions have been implemented on SONY PEG-NX70X.

We explain the implementation using an example of a system execution. An example of a scheduling request and conditions are in the following.

- Request: ClientY needs the support of a meal from 6:30 to 8:00.
- Conditions
  Required qualification: the second grade
  Helpers who have charge of ClientY:
  HelperA, HelperC and HelperD
  Qualification of each helper:
  The second grade: HelperA and HelperC
  The third grade: HelperB and HelperD

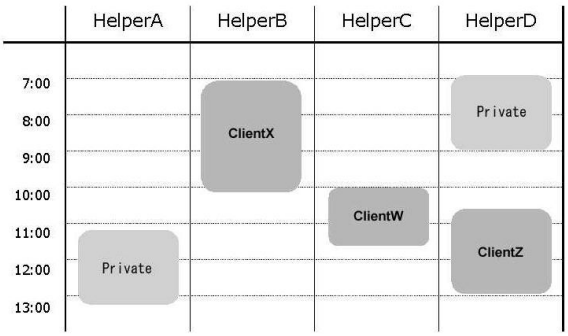

Fig. 3. Helper's schedule.

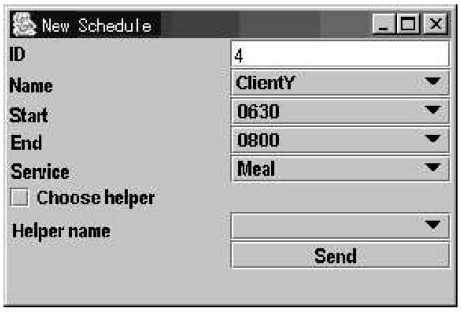

Fig. 4. The registration window for new schedule.

- Total working time (hour)
  HelperA(4), HelperB(6), HelperC(1), HelperD(9)
- Schedules of helpers: the schedules of helpers are shown in Fig. 3. They already have private schedules and care schedules for ClientW, ClientX and ClientZ.

Fig. 4 shows the registration window for a new schedule. A care schedule data is input into this window, and scheduling starts by clicking the button, *Send*. Fig. 5 shows the message sequence chart for scheduling. We explain the scheduling process shown in Fig. 5 in the following.

If the system administrator registers a new schedule, the interface agent sends a *REQUEST* message to the scheduler agent. Then, the scheduler agent starts scheduling. When the scheduler agent receives the *REQUEST* message, it finds a helper who has charge of ClientY and chooses the helpers that

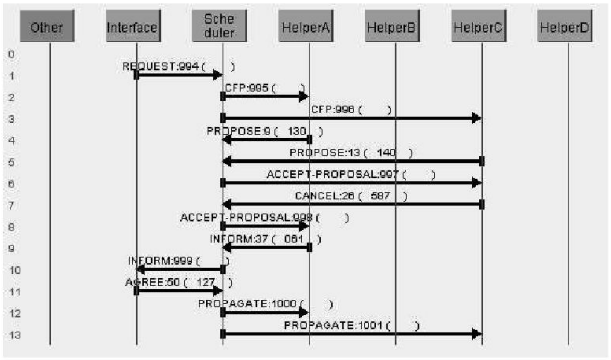

Fig. 5. The message sequence chart for a scheduling.

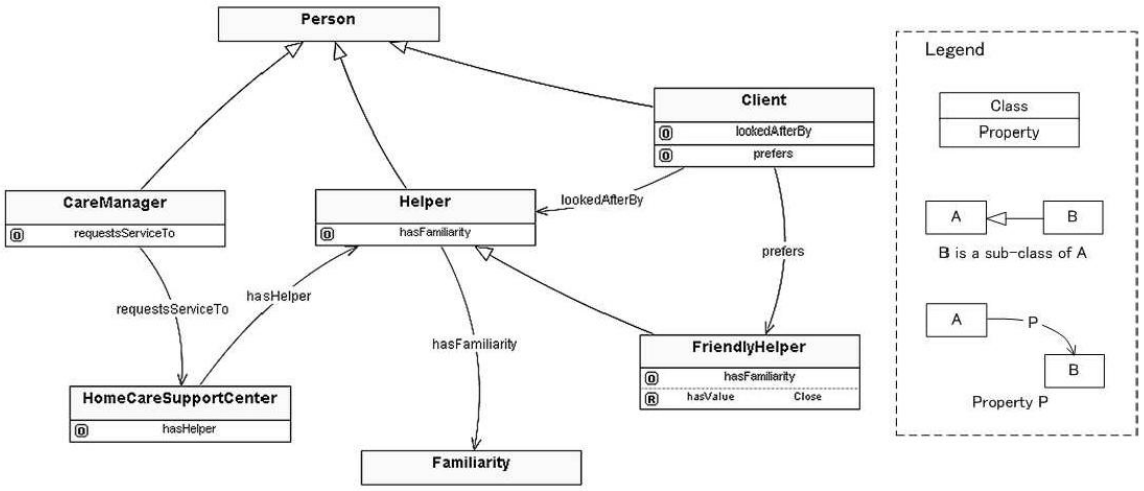

Fig. 7. An example of an ontology for a home care service.

Fig. 6. The screenshot of the PDA application.

have the second grade of the helper qualification. The second grade is required for the support of a meal. In this example, HelperA and HelperC have the second grade. Therefore, the scheduler agent sends *CFP* messages to helper agents of HelperA and HelperC. Then, the both helper agents check owner schedules. Because the schedules of the both helpers are available on the requested schedule, the both helper agents send *PROPOSE* messages to the scheduler agent. When the scheduler agent receives the *PROPOSE* messages, it chooses one helper agent based on a total working time. The scheduler agent chooses a helper agent that has less working time. Therefore, the scheduler agent chooses the helper agent of HelperC and sends an *ACCEPT_PROPOSAL* message. When the helper agent of HelperC receives the *ACCEPT_PROPOSAL* message, it requests HelperC to confirm the requested schedule using PDA. Fig. 6 shows the screenshot of the confirmation window displayed on PDA. In the confirmation window, date (time), a client name and a care content are displayed. The helper can select acceptance or rejection to the displayed schedule by tapping buttons. In this example, we assume that HelperC selects rejection. After rejecting the schedule, the helper agent of HelperC sends a *CANCEL* message to the scheduler agent. The scheduler agent sends the second *ACCEPT_PROPOSAL* message to the helper agent of HelperA, and the helper agent requests HelperA to confirm the schedule. In this example, HelperA selects acceptance. The helper agent of HelperA sends an INFORM message to the scheduler agent. When the scheduler agent receives the *INFORM* message, it sends an *INFORM* message to the interface agent in order to request the confirmation of the schedule. If ClientY selects acceptance, the interface agent sends an *AGREE* message to the scheduler agent. The scheduler agent updates schedules in the database and sends a *PROPAGATE* message to the helper agents of HelperA and HelperC in order to notify the completion of this scheduling process.

In our support system, requested schedules are settled autonomously by agents, and clients and helpers nearly select acceptance or rejection for the schedule. This system reduces the cost of scheduling and improves the efficiency of home care services.

V. Towards an Ontology-based System

Care schedules are decided considering not only vacant hours of helpers but also a helper qualification, the physical and mental condition of a client and the affinity between a client and a helper. Agents in our system should understand these information in order to negotiate schedules more flexibly. As a future work, we have a plan to make an ontology for a home care service using OWL[4]. Fig. 7 shows a simple ontology for a home care service. It represents the overall structure of a concept for a home care service. The ontology described in the OWL abstract syntax is shown in the following.

```
Ontology(
 Class(Familiarity partial)
 Class(HomeCareSupportCenter partial)
 Class(Person partial)
 Class(CareManager partial Person)
 Class(Helper partial Person)
 Class(FriendlyHelper partial Helper
 restriction(hasFamiliarity value
(Close)))
 Class(Client partial Person)
 ObjectProperty(hasFamiliarity
         domain(Helper)
         range(Familiarity))
 ObjectProperty(hasHelper
         domain(HomeCareSupportCenter)
         range(Helper))
 ObjectProperty(lookedAfterBy
         domain(Client)
         range(Helper))
 ObjectProperty(prefers
         domain(Client)
         range(FriendlyHelper))
 ObjectProperty(requestsServiceTo
         domain(CareManager)
         range(HomeCareSupportCenter))
)
```

In this ontology, there is the class `Person`. It has three sub-classes `CareManager`, `Helper` and `Client`. The class `HomeCareSupportCenter` is associated with `CareManager` by the property `requestsServiceTo`. The class `HomeCareSupportCenter` has the property `hasHelper` that associates it with the class `Helper`. The class `Helper` has the property `hasFamiliarity` that represents a familiarity between a helper and a client. The class `Helper` has the sub-class `FriendlyHelper` that has a restriction. The restriction represents that the value (or individual) of the property `hasFamiliarity` has to be `Close`. The class `Client` has the properties `lookedAfterBy` and `prefers`. The property `prefer` means that a client prefers a friendly helper to a usual helper.

The ontology enables agents to consider a familiarity between a client and a helpers. For example, if there exists an individual of the class `Helper` and its value of the property `hasFamiliarity` is `Close`, the type of this individual is inferred to the class `FriendlyHelper`. This inference may be represented with the SWRL abstract syntax[5] as shown bellow.

$$\texttt{Helper(?x)} \wedge \texttt{hasFamiliarity(?x, Close)} \Rightarrow \texttt{FriendlyHelper(?x)}$$

Agents can assign care schedules to the individuals (i.e. helpers) of the class `FriendlyHelper` prior to the individuals of the class `Helper`, because they can see that a client prefers a friendly helper by understanding this ontology. By making a full ontology for a home care service, the negotiation of care schedules can be performed more flexibly, considering various conditions.

VI. Conclusion

We have presented a support system for a home care service. Our support system is designed based on a multi-agent technique in order to settle care schedules autonomously. The clients and helpers decide the care schedules using PDAs, which can be easily handled even by elders. We implemented the scheduling function and the PDA functions. It was shown that the agents in the support system negotiate care schedules autonomously.

As a future work, we have a plan to make a domain ontology and inference rules for a home care service in order to negotiate care schedules more flexibly.

References

[1] Nicholas R. Jennings, "An Agent-based Approach for Building Complex Software System," Commun. ACM, Vol.44, No.4, pp.35–41, 2001.

[2] FIPA,"FIPA Contract Net Interaction Protocol Specification," http://www.fipa.org.

[3] Telecom Italia Lab., "Jade Java Agent Development Framework," http://jade.tilab.com/.

[4] Mike Dean, Dan Connolly, et al., "OWL Web Ontology Language Reference," W3C Working Draft, http://www.w3.org/TR/owl-ref/.

[5] Mike Dean, et al., "SWRL: A Semantic Web Rule Language Combining OWL and RuleML," W3C Working Draft, http://www.w3.org/Submission/SWRL/.