GerAmi: Improving Healthcare Delivery in Geriatric Residences

Juan M. Corchado and Javier Bajo, *University of Salamanca*

Ajith Abraham, *Norwegian University of Science and Technology*

This intelligent environment's key agent, GerAg, dynamically schedules nurses' tasks, reports on their activities, and monitors patient care.

Many countries face an ever-growing need to supply constant care and support for their disabled and elderly populations. Over the past 30 years, the number of Europeans over 60 years of age has risen by about 50 percent, and now represents more than 25 percent of the population. Within 20 years, experts estimate that

this percentage will rise to one-third of the population, or more than 200 million people.[1] Studies in other parts of the world show similar tendencies. In the US, people over 65 years are the fastest-growing segment of the population; by 2020, they'll represent about 1 of 6 citizens. Furthermore, many people over 85 require continuous monitoring and daily care.

Finding more effective ways to provide care for elderly and disabled populations has become a major challenge for the scientific community.[1] Creating secure, unobtrusive, and adaptable environments for monitoring and optimizing healthcare will become vital in the near future. Recently, researchers have begun exploring multiagent systems and architectures to create intelligent supervision systems. These intelligent systems aim to support elderly and Alzheimer patients in all aspects of daily life, predicting potential hazardous situations and delivering physical and cognitive support.

When coupled with RFID, Wi-Fi technologies, and handheld devices, such multiagent systems offer many new possibilities and have given rise to new fields aimed at integrating distributed-intelligence software applications into our daily lives. One such field is ambient intelligence, which

proposes a new interaction model in which developers adapt technology to people and their context[1] and offer users simple, natural, and effortless system interactions.

Ambient intelligence's goal is to develop intelligent, intuitive systems and interfaces that can ubiquitously recognize and respond to user needs. To achieve this, developers must

- provide ubiquitous computation and communication capabilities,
- focus on users' needs during development, and
- create technologically complex environments in medical, domestic, academic, and other contexts.

Guided by these requirements, we've developed Geriatric Ambient Intelligence, an intelligent environment that integrates multiagent systems, mobile devices, RFID, and Wi-Fi technologies to facilitate management and control of geriatric residences. At GerAmi's core is the geriatric agent (GerAg), a deliberative agent that incorporates a case-based planning (CBP) mechanism to optimize work schedules and provide up-to-date patient and facility data. We've successfully implemented a system prototype at a care facility for Alzheimer patients.
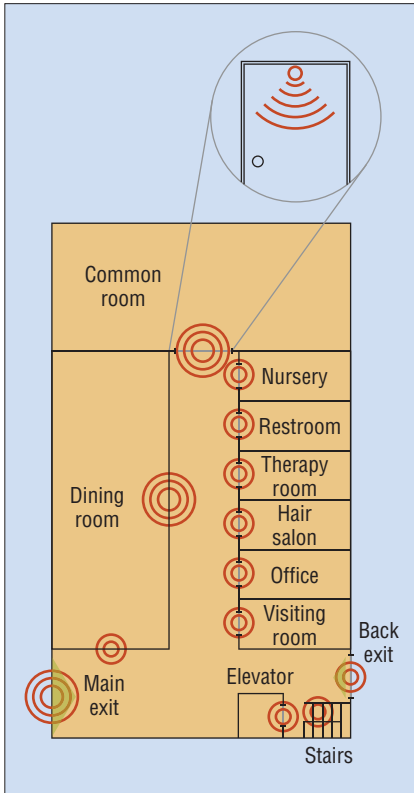
Figure 1. The first floor of Alzheimer Santísima Trinidad Residence of Salamanca. Sensors are positioned above each door and the elevator.

GerAmi overview

To develop GerAmi, we collaborated with the Alzheimer Santísima Trinidad Residence of Salamanca, which was interested in improving its patient services. This collaboration let us both take advantage of the Residence employees' know-how and experiment with our prototype. The Residence is for Alzheimer's patients over 65 years old. Its services and facilities include medical services, including occupational therapy and an infirmary; technical assistance (such as help making a phone call); a terrace and a garden; laundry and tailoring services; a hairdressing salon; a chapel and religious services; social workers; a cafeteria and customized diet planning; and various rooms, including a public geriatric bathroom, a multipurpose room, and separate rooms for reading, socializing, visiting with guests, and watching TV.

The Residence's first floor (see figure 1) contains the main facility rooms; all patient rooms are on the second floor. The residence has a 60-patient capacity, with six nurses, one social worker, and five other
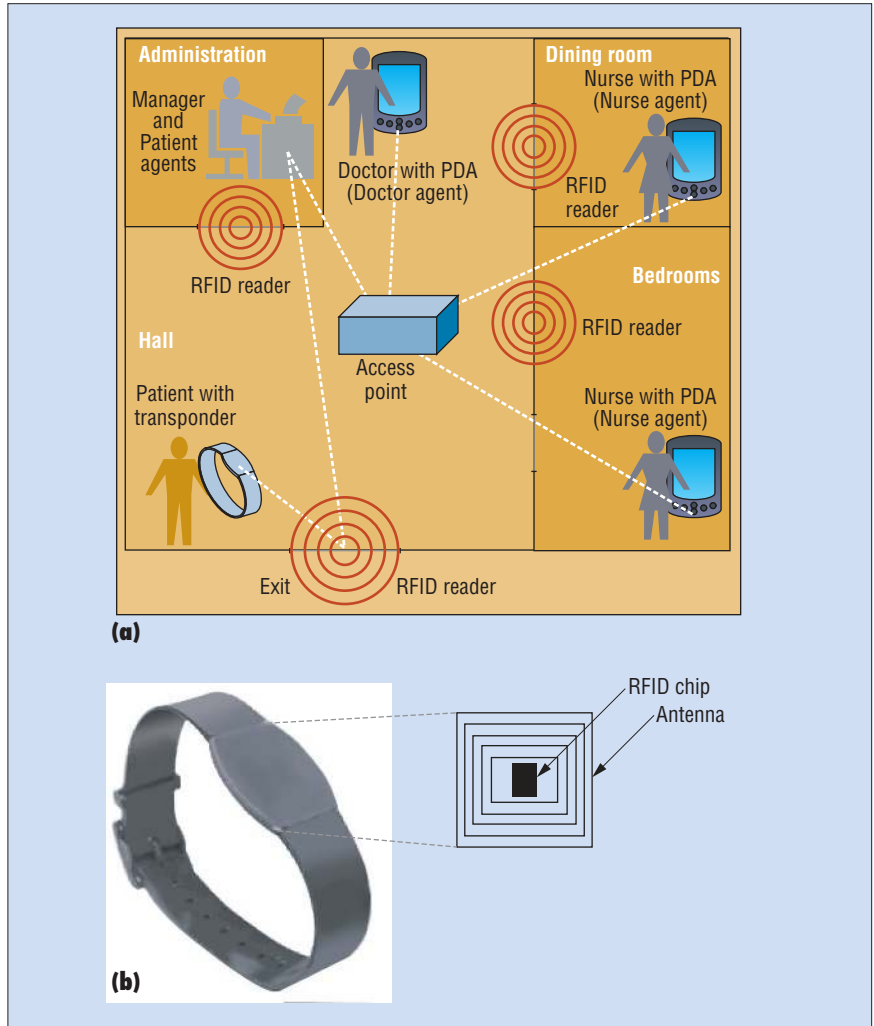
Figure 2. Wireless communications: (a) The Geriatric Residence Multiagent System allows data sharing among the various system components: the transponder bracelets, the door-based readers, and mobile-device-based agents. (b) The transponder bracelets consist of a Hitag S 256 RFID chip and an antenna.

employees with various responsibilities on duty on an average day. We selected 30 patients to test our system.

Basic components

Our system hardware consists of

- 42 ID door readers (Hitag HT RM401 and mobile WorkAbout Pro RFID)—one above each door and the elevator door;
- four controllers—one at each exit, one in the first floor hall, and another in the second floor hall—that have an adjustable capture range of up to two meters; and
- 36 Sokymat ID Band Unique Q5 bracelets—one for each patient and each nurse—that have an antenna and a Hitag S256 RFID chip that issues a 125-KHz signal.

To facilitate data sharing, we developed the wireless, distributed Geriatric Residence Multiagent System (see figure 2a). To track patient location, GR-MAS uses signals from the bracelets (figure 2b), which send their ID numbers to the readers, which forward the data to the controllers. The controllers then send a notification to a system agent that manages and forwards the information to PDAs so the medical staff can readily identify patients' locations. To reduce the solution's complexity, we designed user-friendly interfaces, and we also sought to make GerAmi both robust and easy to use.

The GR-MAS architecture

One reason that agents and multiagent systems (MASs) have become increasingly

relevant for distributed and dynamic intelligent environments is that they support wireless communication. This facilitates system portability to a wide range of mobile devices. Given mobile devices' low memory and processing resources, agents facilitate ubiquitous and transparent interactions and let developers personalize user access. In addition, agents

- are autonomous, reactive, proactive, social, logical, and capable of learning[2];
- can act as an interface between the user and the intelligent environment's elements;
- can adapt to environmental changes or make predictions based on previous knowledge or experience (making them capable of context sensitivity).

Agents are typically integrated into MASs, or agent societies, exchanging information and resolving problems in a distributed way. Such an organization facilitates ubiquitous communication and computation. To model problems, we must therefore achieve an organization-oriented perspective, identifying the roles that each agent plays with the society or system.

After studying the problem requirements, we chose five roles for our GR-MAS model:

- the *Patient* manages patients' personal data and behavior (monitoring, location, daily tasks, and anomalies),
- the *Nurse* schedules the nurse's working day, using dynamic plans based on each assigned patient's needs,
- the *Doctor* treats patients,
- *Security* controls the patients' location and manages locks and alarms, and
- the *Manager* manages the medical record database and doctor-patient and nurse-patient assignments.

System agents

On the basis of these roles, GR-MAS has four different agents: Patient, Doctor, Nurse, and Manager (see figure 2a). Each hour, the Patient agent validates the patient's location, monitors his or her state, and sends a copy of its memory database (patient state, goals, and plans) to the Manager agent to maintain backups. The Patient agent instantiates the patient state at execution time as a set of beliefs; it controls these beliefs through goals that must be achieved or maintained.

We selected beliefs that define a general patient state at this facility, including weight, temperature, blood pressure, food (diet and when to eat), oral medication, parenteral medication, posture change, toileting, personal hygiene, and exercise. Each patient's beliefs and goals depend on the treatment plans and medication that the doctors prescribe. The Patient agent uses the goals to monitor patient state. To determine whether a goal has been achieved, the Patient agent must maintain continuous communication with the other agents. At least once per day (depending on the treatment plan), the Patient agent must contact the Nurse agent. The Patient agent must also communicate periodically with the Doctor agent. Finally,

> Given mobile devices' low memory and processing resources, agents facilitate ubiquitous and transparent interactions and let developers personalize user access.

the Patient agent must ensure that all treatment-indicated actions are carried out.

The Nurse agent is a GerAg that schedules the nurse's working day and manages profiles, tasks, available time, and resources. The plans it generates must guarantee that all the patients assigned to the nurse receive care. Also, nurses can't exceed eight work hours. Every agent generates personalized plans based on the nurse's profile and working habits.

The Doctor agent interacts with the Patient agent to order treatments and receive periodic reports, with the Manager agent to consult medical records and assigned patients, and with the Nurse agent to ascertain the patient's progress. The Doctor agent will eventually also be a GerAg.

The Manager agent plays two roles. First, it plays a security role, controlling the patient's location and managing locks and alarms. Second, it manages the medical-record database and doctor-patient and nurse-patient assignments. Besides pro-

viding security for the patients and medical staff, this agent must ensure efficient assignments among patients, doctors, and nurses.

Robustness and security

The Manager and Patient agents run in a central computer, but the GerAg agent (the Nurse, and, eventually, the Doctor agents), run on mobile devices. To accommodate this, we installed a robust wireless network to extend the existing wired LAN. To ensure failure recovery, we carry out continuous system monitoring. Every agent saves its memory (personal data) onto a database. Patient agents are the most sensitive and thus save their state every hour. When an agent fails, the system easily creates another instance from the latest backup. To ensure that the system's database and server have redundancy and failure recovery, we use a RAID server. If the server fails, the system generates an alarm and automatically prints out all the plans and information that doctors and nurses require to carry out their work. The system also provides secure, authenticated access to patient data. We also use different authorizations for users (logins and passwords) and encrypt messages using a public-key infrastructure and Secure Socket Layer. Moreover, the RFID tags contain only ID numbers, not personal data.

The GerAg autonomous healthcare agent

Currently, we assign each nurse and doctor a GerAg that includes information on patient locations, historical data, and various alarms. Staff members then carry out their duties, following their agent's plan. If they need to modify the initial plan to accommodate delays or alarms, the GerAg can replan in real time.

GerAg works with the high-level concepts of belief, desire, and intention (BDI)[3] and includes a special case-based reasoning system that uses CBP. We therefore refer to GerAg as a CBP-BDI agent[4] that has learning and adaptation capabilities and facilitates work in dynamic environments. It also provides greater autonomy than a pure BDI architecture.[5]

To introduce a CBR motor into a BDI agent, we represent CBR system cases using BDI and implement a CBR cycle. A case is a past experience comprising

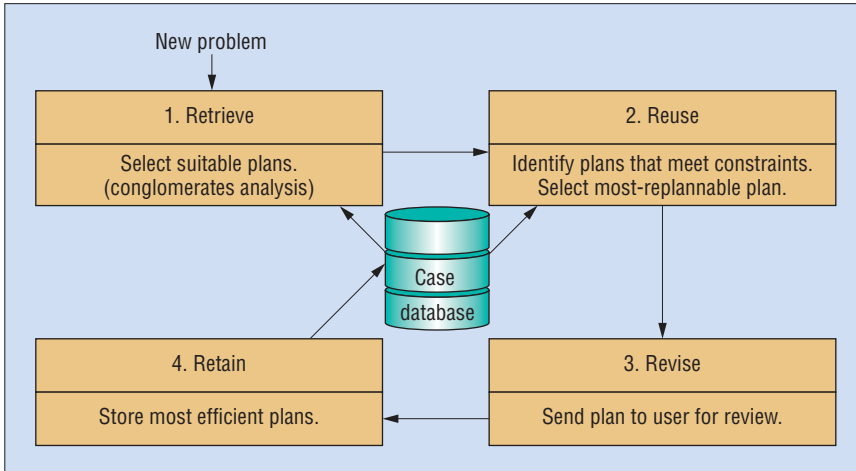- an initial state or problem description

Figure 3. The case-based-planning cycle. CBP's four stages repeat following an interruption, taking previously accomplished tasks into account.

represented as a belief,
- a final state represented as a set of goals (desires), and
- the action sequence—represented as intentions or plans—that lets the user evolve from the initial to the final state.

To obtain CBP, the planning agent's reasoning motor generates plans using cases and planning strategies.

Planning model

Deliberative CBP-BDI agents specialize in generating real-time plans.[4] GerAgs require dynamic planning systems so that they can respond to environmental changes and provide efficient real-time plans to optimize work rosters. Our GerAg's CBP system uses multiple agents to identify daily work plans for each nurse, facilitated by our RFID-based location map of patients, nurses, and doctors.

The GerAg's CBP agent calculates the *most-replannable intention*[6]—that is, a plan that can be easily substituted by another in case it's interrupted. The agent follows the four CBR system stages of retrieval, reuse, review, and retain.[7] In the retrieval stage, the CBP mechanism selects, from the plan database, plans that are similar or contain information similar to the problem case describing the user's aim, restrictions, and profile. To recover similar cases, we use hierarchical, multivariate conglomerates analysis. We chose this method because, compared to other methods, the cases it recovered were more similar to the target problem. Conglomerates analysis considers variables both from a vectorial point and in relation to that vector's components. Basically, it distributes *n* data, of dimension *p*, in conglomerates, or clusters, formed by data that are "similar to each other." To define the conglomerates, it chooses distances and measures of previous similarity using the Mahalanobis distance. The hierarchical-grouping algorithm begins with *n* conglomerates (one for each observation). It then successively unites conglomerates according to similarity.

In the reuse stage, the agent creates a vectorial hyperdimensional space using a B-splines technique to identify all problem restrictions (time, food, equipment, rooms, and so on). It then identifies all possible plans that satisfy a given nurse's requirements. Such plans are geodesic curves (of the vectorial hyperdimensional space), which we can calculate using the Euler theorem.[8] Such geodesic plans guarantee minimum risk and constant efficiency, and satisfy the agent objectives. The geodesic plan with the *minimum Jacobi field* will be the most-replannable intention. The minimum Jacobi field identifies the plan with the most related plans, making it easy to identify a suitable replacement plan as needed.

In the review stage, the nurse reviews the plan. In the retain stage, the system stores plans on the basis of efficiency, which we define as the relationship between the objectives attained and the resources consumed:

$$E_{ff} = \frac{\#(O' \cap O)}{\# R},$$

where # is the cardinal of a set, O' is the set of achieved objectives, O is the set of initial objectives, and R' is the set of consumed resources.

Figure 3 shows the CBP system's four stages. When an interruption occurs, the system initiates a new CBP cycle, taking into account its previously accomplished tasks. That is, in the new retrieval stage, it recovers plans with a problem description similar to the current situation. The most-replannable intention guarantees that it will recover at least a few plans closest to the initial geodesic plan (the remaining plans are invalid owing to restrictions, previously accomplished tasks, and so on) as well as new plans.

GerAg agents in operation

The GerAg schedules workdays using dynamic plans based on tasks related to each assigned patient. Given a set of beliefs B compatible with an initial problem E, the GerAg can generate a plan database containing all possible plans produced by combining compatible beliefs. The GerAg's available beliefs are tasks, resources, and time.

A task is a Java object that contains data on the patient who needs service, the service description, and the task's time limits. For each task, the GerAg establishes one or more goals aimed at eventually achieving the entire task. A goal—also a Java object—identifies what the GerAg wants to achieve (completing a task) and under which conditions (restrictions). A goal might contain parameters and define creation conditions (which let the agent define the conditions for achieving the goal), context conditions (conditions that must be fulfilled), and drop conditions (which cannot be fulfilled). To achieve its objectives, each goal triggers plans, which are procedures written in Java code. A goal can create new goals (subgoals) to achieve its objectives. For example, for the rehabilitation task, the GerAg creates a new goal for each of the task's concrete exercises.

The CBP system constructs plans as a task sequence that nurses must carry out. It forms a problem description using the required tasks, available resources, and the nurse's work schedule. In the retrieval stage, it recovers—from the beliefs database—descriptions of similar problems using various similar algorithms (cosine, clustering, and so on). In our case, we permit a tolerance of 20 percent.

Once the system recovers the most similar problem descriptions, it moves to the reuse stage to recover the associated solutions.
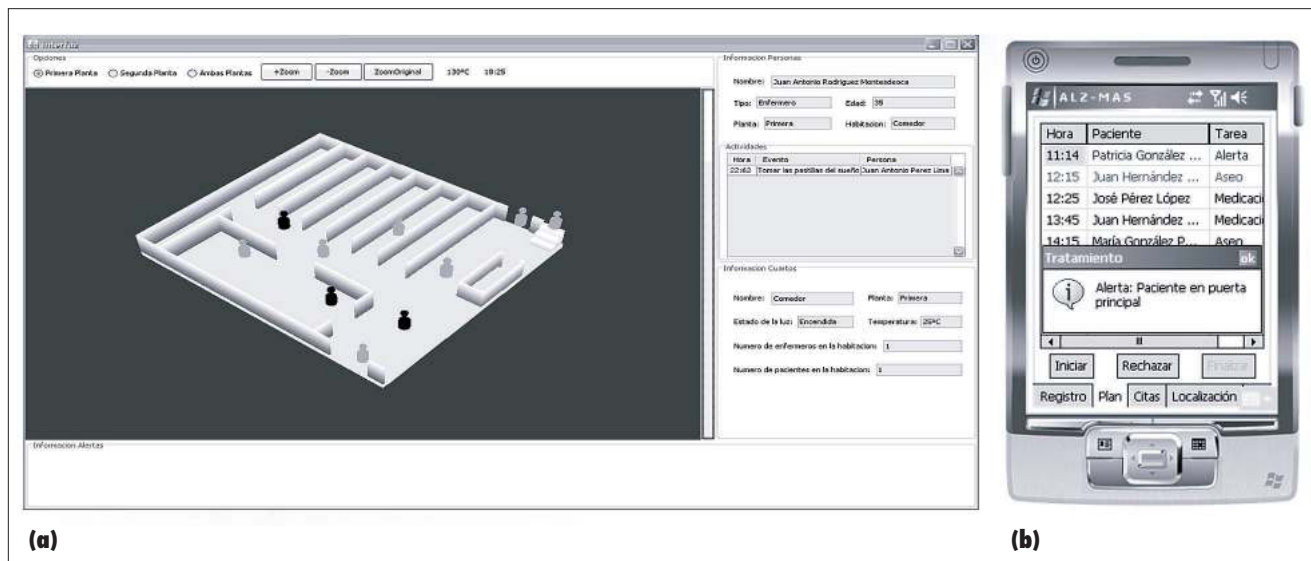
Figure 4. Task planning and assignment. (a) The Manager agent's interface uses a 2D reference system and RFID devices to identify patient location. (b) A cell-phone screenshot shows a nurse's current plan, offering a list of patients, their needs, and the schedule for meeting those needs. Clicking on a patient's name opens a window that shows the patient's status.

A solution contains all the plans previously used to achieve the GerAg's objectives for a particular problem description (assuming that replanning is possible), along with the solutions' efficiency. The GerAg then combines the recovered solutions to construct a plan. Currently, the GerAg controls plan processing (scheduling). We focused its replanning mechanism on each task's objectives and resources, as well as on the nurse's objectives and the resources available to carry out the global plan. Each nurse's global plans are to attend to the patients and to not work more than eight hours. The time available is thus a problem restriction that influences the restrictions' hyperplan. The resources required for some tasks include food, equipment, and rooms.

Planning-task examples

Because nurses vary according to qualifications and routine tasks—and some perform tasks with greater skill or carry them out more quickly—the GerAg must manage the nurses' profiles. It also maintains a close relationship with the Manager agent, which assigns the nurses and doctors to patients using its own CBR reasoning motor.

When the Manager agent assigns a new task to a nurse, it examines the available nurses' profiles and the current situation's needs and then allocates the task. For example, not all nurses are equally qualified for rehabilitation. The Manager agent will thus give the most qualified nurses the patients with the greatest rehabilitation needs, taking into account their eight-hour work limits and each rehabilitation task's time. The Manager agent also accounts for improvement rates in rehabilitating patients, the arrival of new patients, holiday work rosters, and so on.

Second, the Manager agent stores, in the beliefs database, each task's required time, described as

$$t_j = \underset{j,k}{\text{Max}} \left\{ t_{jk}^i \right\}$$

where j indicates the type of task, k is the nurse with the most suitable profile to carry it out, and i is the patient requiring the task.

Once the Manager agent completes its task assignments, it communicates each task assignment to the corresponding GerAg. At this point, planning begins. The GerAg first accounts for the nurse's available time and the task's required time, as well as available resources and patient location. It then recovers and combines similar previous plans and solutions from the beliefs database.[6]

Many measurements are required to standardize the time taken to arrive at a given room or to move a patient from one room to another (which depends on the patient's dependency level). The GerAg includes such times directly in the time it assigns for each task. Patient location significantly influences decisions about whether to interrupt a plan. If, for example, a nurse is must take a patient's dinner to a given room but finds that the patient is in a different room, the plan must be interrupted.

As figure 4a shows, we define patient location using a 2D reference system and RFID devices, which let us rapidly assess replanning needs. Figure 4b shows an example plan, viewed through the target nurse's PDA interface.

The system might interrupt a plan for many reasons. In this facility, those reasons include an emergency situation, a resource failure, a patient crisis, or visitors who are unexpected or stay too long. In such a situation, the GerAg rejects the initial plan and seeks an alternative. It first changes the task order, attempting to maintain the assignment originally allocated by the Manager agent. The new plan must meet the initial objectives; if this isn't possible, nurses must be reassigned in a way that keeps changes to a minimum. Reassignment requires that the GerAg account for all existing nurse assignments, pending and completed tasks, and nurse profiles so that it can prioritize according to the target task. The GerAg must then replan the target nurse's tasks; assuming the replanning is positive (that is, it's possible to complete both the existing tasks and the new task), the process is complete. If the replanning is negative, the GerAg replans with the next nurse in the ranking. Finally, the GerAg stores the plan and its efficiency level in the beliefs database.
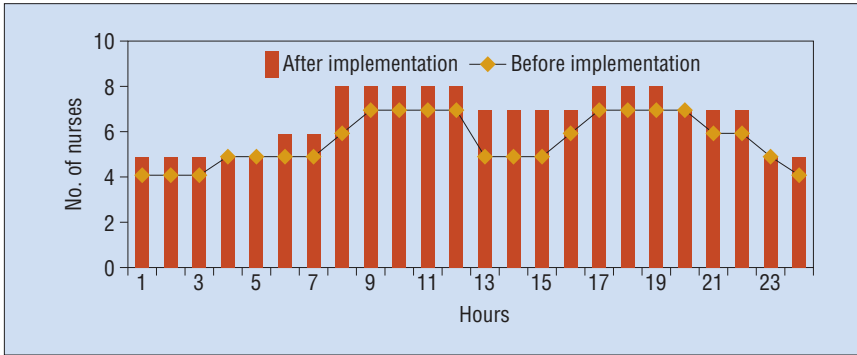
Figure 5. The average number of nurses working simultaneously. As the graph shows, after we implemented GerAmi, the facility needed fewer nurses on the clock at any given time.

Experience and results

We tested GerAmi over a three-month period. During this period, we evaluated the system's usefulness from different viewpoints. Figure 5 shows the average number of nurses working simultaneously during each 24-hour period before and after we implemented the prototype in January 2007 (we collected data from October 2006 through March 2007). The average number of patients was the same before and after the implementation.

To test the system, we instantiated 30 Patient agents, 10 Nurse agents, 2 Doctor agents, and 1 Manager agent. Because we've yet to incorporate planning capabilities in the Doctor agents, we focused on the Nurse agents. As the figure shows, GerAmi gives nurses more free time, which they can then dedicate to caring for special patients, exercising patients or helping them with leisure activities, enhancing their own knowledge, and simply talking with the patients and their families. GerAmi also substantially reduced supervision and control-task time, as well as time spent attending to false

alarms. It also increased time for direct patient care.

Task-time reductions

We divide nursing tasks into two categories. *Direct-action tasks* require the nurse to act directly on the patient throughout the task. Examples here include administering medication, toileting, repositioning, and feeding. During *indirect-action tasks*, the nurse doesn't act directly on the patient throughout the task. Examples here include filling out reports, monitoring patients, and periodic visits. Because GerAgs take on some indirect actions, nurses can dedicate more time to personal patient care.

In our testing, we analyzed data according to the two tasks. We first analyzed the problem and collected data. We then obtained the average time nurses spent carrying out their duties with a given patient, accounting for patient type and dependency level, and the nurse's professional level. Table 1 shows the times for direct-action tasks; dependency level 2 indicates patients with more severe conditions or diseases.

Because GerAmi should decrease indirect-task times, we were especially interested in the results in this area. As Table 2 shows, GerAmi significantly reduced the time spent on all indirect tasks.

Interruptions: Measuring learning capacity

Traditional BDI (deliberative) architectures lack learning capabilities. In contrast, our GerAg agent improves its knowledge using the CBP system. This noticeably reduces interruptions for replanning. It also reduces the gap between BDI agent formalization and implementation.[4]

To evaluate GerAgs' learning capacity, we measured plan quality. The number of interruptions indicates the number of replannings required to complete a plan. To learn, the GerAg executes CBP cycles. Our results show that, after the GerAg executed 100 plans, the average number of interruptions decreased 30 percent. Specifically, its average went from 9 interruptions per day after executing 10 plans to 8 interruptions after executing 50 plans to 7 interruptions after executing 100 plans. So, we conclude that the GerAg improves its behavior with learning and that, on average, the number of interruptions won't decrease to fewer than 7 per day.

Patient care and facility security

In designing and developing GerAmi, we focused on the patients' perspective and on the relationship between the patients and staff. Because our dynamic planning mechanism permits real-time task replanning, patient care improves. GerAmi also facilitates the more flexible work-shift assignments: because workers spend less time on routine tasks, they have more time to pur-

Table 1. Average direct-task times (in minutes) using GerAmi (Geriatric Ambient Intelligence) for patients with varying dependency levels.

Dependency degree	Cleaning	Feeding	Oral medication	Parenteral medication	Posture change	Toileting	Exercise	Other tasks
1	35	18	8	30	25	8	60	10
2	45	28	11	42	50	30	90	10

Table 2. Average indirect-task times (minutes) before and after GerAmi.

	Monitoring	Reports	Visits	Other	Total
Before GerAmi	167	48	73	82	370
With GerAmi	105	40	45	60	250

sue extra activities. All work is automatically monitored, as are patients' activities. Residence managers can analyze the resulting information using knowledge discovery techniques, to improve the patients' quality of life and the facility's efficiency.

GerAmi also improves the Residence's security by

- monitoring patients and guaranteeing that each is in the right place,
- ensuring that only authorized personnel can access protected areas, and
- storing information more securely, using redundancy and generating continuous backups.

Furthermore, it protects information access to guarantee confidentiality.

As our results show, the GerAmi system creates a distributed, intelligent environment that helps healthcare facilities and providers contend with the increasing challenges of caring for Alzheimer's patients, the elderly, and people with other disabilities. We did have a few issues in implementing the system, partly because the nurses and other facility workers weren't familiar with PDAs. To address this, we offered training to introduce the technologies and teach employees how to use the system interface. After that, our primary difficulties were in installing the wireless access points (we faced some signal-propagation problems) and collocating RFID door readers. Otherwise, the system runs smoothly, with only minor problems. □

References

1. L. Camarinha-Matos and H. Afsarmanesh, "Design of a Virtual Community Infrastructure for Elderly Care," *Proc. 3rd IFIP Working Conf. Infrastructures for Virtual Enterprises*, Kluwer, 2002, p. 635.
2. M. Wooldridge and N.R. Jennings, "Agent Theories, Architectures, and Languages: A Survey," M. Wooldridge and R. Jennings, eds., *Intelligent Agents*, Springer, 1995, pp. 1–22.
3. M.E. Bratman, *Intentions, Plans and Practical Reason,* Harvard Univ. Press, 1987.
4. J. Bajo et al., *Hybrid Architecture for a Reasoning Planner Agent*, LNAI 4693, Springer, 2007, pp. 461–468.
5. J.M. Corchado and R. Laza, "Constructing Deliberative Agents with Case-Based Reasoning Technology," *Int'l J. Intelligent Systems*, vol. 18, no. 12, 2003, pp. 1227–1241.
6. M. Glez-Bedia and J.M. Corchado, "A Planning Strategy Based on Variational Calculus for Deliberative Agents," *Computing and Information Systems J.,* vol. 10, no. 1, 2002, pp. 2–14.
7. A. Aamodt and E. Plaza, "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches," *AI Comm.*, vol. 7, no. 1, 1994, pp 39–59.
8. R.E. Bellman, *Dynamic Programming*, Princeton Univ. Press, 1957.

The Authors

Juan M. Corchado is an associate professor at the University of Salamanca, where he directs the Intelligent Information System Group (http://bisite.usal.es) and the MSc programs in computer science. He has also been a research collaborator with the Plymouth Marine Laboratory since 1993. His research interests include agents and multiagent systems, ambient intelligence, case-based reasoning, and real-time forecasting. He received a PhD in computer science from the University of Salamanca and a PhD in AI from the University of Paisley. Contact him at Departamento de Informática y Automática, Facultad de Ciencias, Univ. de Salamanca, Plaza de la Merced S/N, 37008, Salamanca, Spain; corchado@usal.es.

Javier Bajo is an associate professor at the Pontifical University of Salamanca. His research interests include multiagent systems, decision-support systems, and intelligent environments. He received his PhD in computer science from the University of Salamanca. Contact him at Escuela Universitaria de Informática, Universidad Pontificia de Salamanca, Compañía 5, 37002, Salamanca, Spain; jbajope@upsa.es.

Ajith Abraham is a visiting professor at the Norwegian University of Science and Technology. His research interests are in computational intelligence, focusing on applications such as network security, Web services, Web intelligence, multicriteria decision-making, and data mining. He received his PhD in computer science from Monash University, Melbourne. He's a senior member of the IEEE. Contact him at the Norwegian Center of Excellence, Center of Excellence for Quantifiable Quality of Service, Norwegian Univ. of Science and Technology, O.S. Bragstads plass 2E, N-7491 Trondheim, Norway; ajith.abraham@ieee.org; www.softcomputing.net.