

RFID technology and AI techniques for People Location, Orientation and Guiding

M. D. R-Moreno[†], B. Castaño[‡], M. Carbajo[†] and A. Moreno[†]

[†]Departamento de Automática. Universidad de Alcalá. Madrid, Spain.
{mdolores,mcm, angel}@aut.uah.es

[‡]Departamento de Matemáticas. Universidad de Alcalá. Madrid, Spain.
bonifacio.castano@uah.es

Abstract. One of the main problems that we have to face when visiting public or official buildings (i.e hospitals or public administrations) is the lack of information and signs that can guide us. Thanks to the new technology advances, the electronic communication networks can be focused on an objective environment. These techniques can be used to help users to get their right location and orientation.

This is the framework we are chosen in this article. The solution proposed in this paper uses a detection and a location system based on wireless technology and Artificial Intelligence (AI) techniques to plan and inform about the paths the users can follow. The AI system is called PIPSS and integrates planning techniques and scheduling methods.

Keywords: AI Planning & Scheduling, Montecarlo method, RFID, Location & Orientation.

1 Introduction

The growth and development of information and communication technologies (ICTs) has opened a new range of applications, specially in the environment where we live. Places such as houses, offices or public institutions will be able to recognize us and to adapt to our taste, needs and preferences as soon as we come inside them. Although this can seem taken from a science fiction movie, there are already some academic and industrial initiatives that begin to show the strong current tendency to incorporate this type of techniques.

In this paper we present a solution to the problem that a user has to face when visiting large surfaces of high affluence level of people. The application is called SIGUEME (Sistema Inteligente de Guiado para Entornos Multiusuario Extensos - Intelligent Monitoring System in Big Multiuser Enviroments)¹ and it includes software and hardware elements: RFID passive technology [4] for the user detection, and an AI planning and scheduling techniques for the orientation and the guiding of the patients. Radio-frequency identification (RFID) is an

¹ In English SIGUEME means Follow Me

automatic identification method, relying on storing and remotely retrieving data using devices called RFID tags or transponders. An RFID tag is an object that can be applied to or incorporated into a product, animal, or person for the purpose of identification using radio waves. RFID tags come in three general varieties: passive (require no internal battery), active (require power supply) or semi-passive (are battery-assisted) tags. In our application we have used the first type.

We have chosen a concrete example of a big surface with people going back and forward to test our system: a medical centre. The scenario is as follows. At the information desk the receptionist identifies the patient, verifies the appointment and give him an individual RFID card. Each time he goes through a RFID arch, the system detects him and calculates his position. The information to guide the patient is displayed on several screens located at geographically strategic points along the building. When looking for the place to go, the visitor can lose his way, in that case, the system will detect this situation and will correct the mistake.

The paper is structured as follows. Section 2 describes the philosophy we have followed to place the RFID sensors. In section 3 the architecture of our system is presented. Then, experimental results are shown. Finally, conclusions are outlined.

2 The Building Description

In order to follow the patients' movements inside a building, we have used a set of RFID detectors placed at strategic points. Once the RFID detectors are placed, the building gets divided into zones (rooms between the detectors). The zones are not all equal and they can have different features and functions. We have considered three different zones:

- Input Zones: are the first zones reached by the patients when they get into the hospital. They are outside any RFID detector. The information desk is placed in these zones and generally there is only one.
- Transition Zones: are intermediate zones between two or more RFID detectors. They contain the information screens.
- Destination Zones: represent the waiting rooms. Besides, when a user is on his way out, the input zones play the role of destination ones.

All of these zones are enclosed by RFID detectors in such a way that the user entrance into any zone and his subsequent exit are recorded the whole time. Our system will manage all data and it will control the situation, movements and instructions for each patient. There is not limitation on the amount of RFID cards the application can handle.

Once the building is divided into zones, we can describe it by a **graph**. In this graph, zones are the vertexes and detectors are the edges. That is, two vertices/zones are connected if there is a RFID arch between them. The graph is an undirected one and it could not be simple. We use the adjacency matrix

for storing this graph in the system. But, instead of using the number of edges connecting two different vertexes, which is usually one, we use the RFID detector number between the corresponding zones.

The Figure 1a shows a sketch of this scale building layout with the detectors and message monitors. The Figure 1b shows it corresponding matrix.

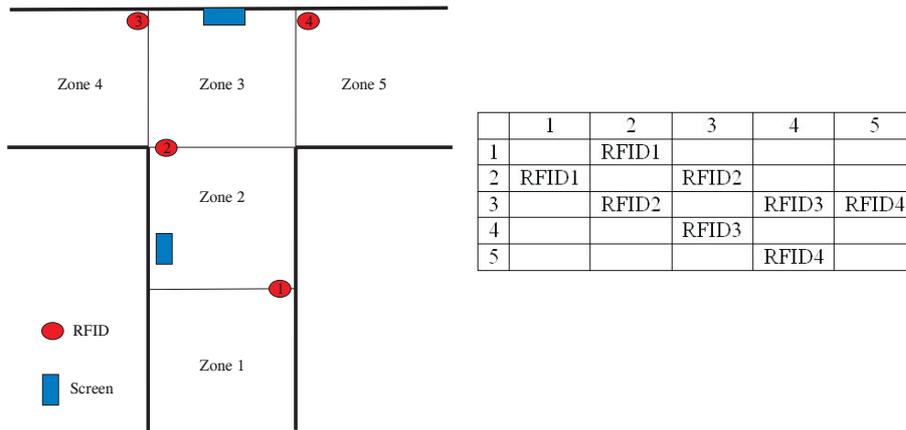


Fig. 1. a) The building layout. b)The adjacent matrix.

3 The Architecture

In this section we describe all the elements of the SIGUEME system [2]. It is composed of the following subsystems:

- The Control Subsystem manages and controls the information and communication with the rest of the subsystems.
- The Detector Subsystem is in charge of detecting the users along the building by means of the RFID cards.
- The Reasoner Subsystem guides the user inside the building.
- The Information Subsystem allocates the information needed for the other subsystems and the users.
- The Visualization Subsystem represents and visualizes the guiding information in the corresponding screens.

3.1 The Control Subsystem

The Control Subsystem is in charge of taking the control of the other subsystems. Besides, it has to control the every day patient data and prepare a daily doctor's

appointments list. It also knows the RFID available cards and the ones in use at every moment. This subsystem is operated from the information desktop. At the beginning of the day, this subsystem creates a list with all the patients with the appointments for that day. When the visitor arrives to the admission desk, the employee identifies and introduces him in the system. Simultaneously, a RFID card with a unique number is given to him. The card number is also notified to this subsystem and it identifies unambiguously its owner inside the building. The patient must keep this card the whole time inside the building.

At the same time, this subsystem finds out the patient destination and asks the Reasoner Subsystem for the track, across the hospital, for the patient. When the Reasoner Subsystem gives its answer, the Control Subsystem records it and it takes care of guiding the person towards its destination.

To deal with this task the Control Subsystem registers and distinguishes each one of the patient's RFID card. This strategy provides our application with the information about the location of all the persons inside the hospital, and even what their movements are. When a user goes through a RFID detector, the Control Subsystem calculates if he is in the right location. If a patient loses his track, the subsystem recognizes the situation and calls the Reasoner Subsystem to generate a new plan. When a person arrives to the destination point, this subsystem finishes its orientation job. If the patient leaves the hospital, he must return the RFID card and the employee will take him out of the system.

3.2 The Detector Subsystem

The aim of this subsystem is to detect the persons moving inside a building. It consists of a group of RFID detectors placed at several points in the building.

These detectors give a signal when a patient with a RFID card passes close to them. This signal is decoded and sent to the Control Subsystem.

In this work we have developed two different detector subsystems. The first one is a hardware prototype that scales a building with one input zone, two transition zones and two destination ones. This scale model uses four RFID short range detectors and is connected to the main computer by RS232/RS485 port. This allows us to set the detectors a thousand meters away from the Control Subsystem. Reproducing the patient's movements inside the building is as easy as passing the RFID cards over the small detectors and recording the signal produced.

Although this strategy reproduces very faithfully the detection process, it is not useful at all if we want a statistically significant amount of data for testing our system. With this aim we have developed a Montecarlo simulation program that generates patient movements across the RFID detectors and works in the same way that the scale hardware model does. It has even the advantage that it can be adapted to a new building topology in a faster and easier way than the hardware prototype. Section 4.1 describes the simulation process.

3.3 The Reasoner Subsystem

This subsystem is based on AI Planning and Scheduling techniques. The Information Subsystems translates all the patient and building information into a suitable format for input to the Reasoner Subsystem. That is, the initial zone, the RFID id given to the patient, the target zone and the connections between the different parts of the hospital. So, if a person loses his way, the system is able to detect his position. At that moment, the Reasoner Subsystem is called and a new plan is generated for the lost patient.

This new plan will be translated in new screen messages managed by the Visualization Subsystem to guide the patient. The Reasoner Subsystem will produce as many times *guiding* plans as needed until each patient achieves his goal location. Finally, when a patient has accomplished his whole medical schedule, the Reasoner Subsystem will generate a new *guiding* plan to guide him right to the exit.

This subsystem is composed of PIPSS (Parallel Integrated Planning and Scheduling System) [9]. It is a system that integrates AI planning and scheduling techniques. It is based on HPP (Heuristic Progressive Planner) [10] and scheduling algorithms [3]. Its open architecture using object oriented interfaces allows the implementation and execution of different planning algorithms, scheduling methods and planning and scheduling integration schemes.

PIPSS has two kinds of planning searches: enforced hill-climbing and greedy best-first search. One type of scheduling algorithm called ISES [3] or the possibility to disable scheduling. And also, two types of planning and scheduling integration schemes: scheduling after planning or scheduling inside planning.

In addition to this, a thread search has to be instantiated with one of the three kinds of operators sets or vectors that heritages from HPP. Each one in sequence or in parallel can be launched in threads. These vectors are:

- *A vector* contains all the possible instantiated operators in the problem.
- *B vector* contains the instantiated operators generated thanks to the relaxed GraphPlan heuristic used for computing the costs in FF [8]. Each operator is relaxed by simply eliminating its delete list. The relaxed plan graph is similar to that produced by the GraphPlan, except that it does not contain any mutual exclusion relations. All the operators used in the different action levels will be collected and saved in the *B vector*.
- *C vector* is generated using an additive heuristic h_{add} for computing the heuristic cost as in HSP [1]. Considering the subgoal independence under this heuristic, *C vector* will contain all the operators that are part of the relaxed plan.

For example, if we consider the blocksworld domain with 4 possible operators (pick-up ?, putdown ?, stack ? ?, unstack ? ?); and the problem has 3 blocks (A,B,C) where blocks A and B are on the table, and block C is on top of block A, then the number of possible applicable operators in the *A vector* is 24 (this value is obtained by substituting each variable in the operators by the objects in the problem). At a first glance, we can see that there are 6 unnecessary combinations,

that is, when the instantiated variables in the *unstack* and *stack* operators are equal (i.e. (*unstack A A*), (*unstack B B*), etc). Table 1 shows the elements of this vector.

Table 1. *A Vector* for the blocksworld domain problem.

pick-up A	pick-up B	pick-up C	putdown A	putdown B	unstack C B
putdown C	stack A B	stack B A	stack A C	stack C A	unstack B C
stack B C	stack C B	unstack A B	unstack B A	unstack A C	unstack C A

Then, the *B vector* will contain 14 operators as Table 2 shows. The relaxed plan graph consists of four fact layers and three action layers.

Table 2. *B Vector* for the blocksworld domain problem.

pick-up A	pick-up B	pick-up C	putdown A	putdown B	unstack C B
putdown C	stack A B	stack B A	X	stack C A	unstack B C
stack B C	stack C B	X	X	X	unstack C A

However, the *C vector* only contains **9 operators**

Table 3. *C Vector* for the blocksworld domain problem.

pick-up A	pick-up B	pick-up C	putdown A	putdown B	X
putdown C	stack A B	X	X	X	X
stack B C	X	X	X	X	unstack C A

PIPSS uses PDDL [5] as the defining language for its domain and problem input files. However, PDDL lacks some characteristics for expressing temporal and multicapacity resources constraints that PIPSS can handle. Thus, it has been necessary to extend PDDL from its standard basis. The extensions have been based on the way PIPSS deals with these issues and how it describes them with its domains and problems definition language. So if a patient has different appointments in the same day, PIPSS can schedule them.

3.4 The Information Subsystem

This subsystem is in charge of keeping the whole information of the patients, the destination goals and the building information in the format understandable for each subsystem.

It uses two data sources: the building description and the patients information. It is very important that these data can be given by means of external files (although it can be introduced/modified by hand to the system), because it makes possible to change either the building or the patients information without modifying the software.

The first file basically consists of the graph building adjacent matrix, where all the zones and detectors are recorded.

The second one consists of a file with all the information about patients and their doctor's appointments. The application allows us to import this file with a predefined structured that will be saved in the database.

This subsystem also saves the following data: every signal produced by the RFID detectors, the doctor appointments for each patient, the *guiding* plans for each patient by the Reasoner Subsystem and the messages given by the Visualization Subsystem. This way we can later on analyze the system behavior and find deficiencies on the messages sent to the users, in the localization of the detectors inside the building, etc since our goal is to have a system easy to use by the user.

3.5 The Visualization Subsystem

It is in charge of sending the corresponding guiding information to the screens where the user is close to. It must be able to send the right messages to the correct places to help all the visitors simultaneously.

It is composed of a set of screens located at the building transition zones. In these devices appear the appropriate information for guiding the patients within that zone. The messages in the screen show to each user the direction he has to follow according to the *guiding* plan generated by the Reasoner Subsystem. The information that the patient visualizes is: the patient's id and the direction he has to follow. In our prototype we have chosen four different instructions: 'straight', 'turn right', 'turn left' and 'go backwards', followed by the corresponding arrow. Any other indications for the same patient will be erased from any previous display devices.

4 Experimental Results

In this section we describe the simulator developed to test the SIGUEME architecture, and the results obtained by PIPSS in the hospital domain when it is compared to other planners.

4.1 The simulator

It is a program that simulates the patient movement through the different RFID detectors located in a concrete set of places inside a building. It is based on the Montecarlo method [11] and it only needs to know how the building zones, that is, the building graph adjacent matrix.

The purpose is to exactly provide the same detection data the hardware RFID device would supply. Doing it that way, we can easily collect a statistical significant amount of patient interactions. Software simulation allows us to avoid errors that could occur by hand simulation such as to forget any RFID card during the experimentation or to go across two different not contiguous detectors. In this first approximation we have presumed that there are no lost detections. The application also assumes the simplification that the patients, moving inside

the medical centre, are able to find and understand the messages supplied by the information screens without demanding any other kind of assistance. Nevertheless, there is a certain chance that the person loses the information provided by the monitors and it had a lack of orientation that, in some cases, could get it out of the planned track. This situation can be experimented by the simulation with several levels of disorientation probabilities.

The simulator takes into account two different orientation sources for a person moving inside a building. The first one is given by the building layout and we call it "geographical orientation". This means that if a person is walking along a corridor there is a bigger probability to follow in the same direction than to go backward. In the same way, when he reaches an intersection there will be a different probability for each possible way the visitor could take according to his movement. To deal with the geographical source we have assumed a fixed probability distribution for each building zone that has two or more contiguous ones. This distribution shows the probabilities to reach each one of the contiguous ones. The second source is the set of the information screens. We can easily admit that a patient has a certain probability of misunderstand the information the system provides to him. Obviously, this situation will depend on the particular person and the monitor position. However, in our first model we have assumed that all the patients in all zones have the same probability P of getting the right direction. The two information sources are linearly combined in such a way that when $P=0$ we only have the geographical probability distribution and if $P=100$ the patient never loses his track. The visitor movements are simulated applying the Montecarlo method on that probability distribution.

The simulation program will work on demand from the Control Subsystem, until all the patients leave the medical centre.

4.2 Results

The Hospital domain is a simple path planning domain where people move among the different parts of a building. Then, the building graph adjacent matrix is used to set the problems. The action of moving from one room to another is durative, so all the people can independently move in parallel.

Four planners have been tested against PIPSS. We have also used four different PIPSS settings, so it can be said that there is a total of eight planners. They are explained as follows:

- PIPSS-A: PIPSS executing *A vector*, enforced hill-climbing, ISES and sequential search.
- PIPSS-B: PIPSS executing *B vector*, greedy best-first search, ISES and integrated search.
- PIPSS-C: PIPSS executing *C vector*, greedy best-first search, ISES and sequential search.
- PIPSS-ABC: PIPSS running three threads like the previous three configurations.
- LPG-speed [6]: non-deterministic planner LPG trying to achieve a solution as fast as possible.

- LPG-quality: non-deterministic planner LPG trying to achieve a solution with the lowest makespan (this modality cannot be launched more than once to get a better solution).
- CPT1 [12]: planning system for optimal temporal STRIPS planning with a distinguished Performance in Optimal Planning (Temporal Domains) at IPC'06.
- CRIKEY [7]: a temporal planner written in java.

We have generated 20 problems, increasing the number of persons and rooms. All executions have had a maximum available time to find a problem of ten minutes and all planners have been launched under Windows XP.

All PIPSS modalities and LPG speed found a solution to all the problems, LPG quality solved 75% of them, CPT solved 50% and, finally, CRIKEY solved 45%.

In this domain, PIPSS is the best performer (in time and makespan). All PIPSS executions find 100% of the solutions along with LPG speed, but PIPSS makespans are much better by far. LPG quality only solves 75% of the problems and its makespans are never better than those of PIPSS A or PIPSS B. The fastest searches seem to be achieved by PIPSS A and PISS ABC, both launching an EHC thread (however, CPT is a bit faster for the first ten problems, which are the only ones that it solves). The reason why PIPSS obtains the best makespans in this domain is that the EHC implementation that it uses -which comes from HPP- is very good at providing solutions with fewer steps. Since the duration of all the actions of this domain is the same (one unity of time) and since people move independently (so moving a person can be seen as a single subproblem), this means that shorter solutions will result in lower makespans. Of course, in order to do so, it is also important that the scheduler does a good job, which means that ISES performs very well when it is provided with an adequate plan, as in this case. CPT and CRIKEY only find 50% and 45% of the problems respectively but both provide good makespan, although CRIKEY is too slow.

5 Conclusions

In this paper we have presented SIGUEME, an application based on RFID and AI Planning and Scheduling techniques to solve the problem of guiding persons through large surfaces of high affluence level of people. A specific system called PIPSS has been developed for this purpose. We have tested against other state of the art planners, and PIPSS gets the best performance in makespan and time.

This system has been implemented satisfactorily in a scale prototype that has successfully proved its viability and good performance. A simulator based on the Montecarlo method has been used to test the whole architecture.

Acknowledgments

This work has been funded by the Junta de Comunidades de Castilla-La mancha project PAI07-0054-4397.

References

1. BONET, B., AND GEFFNER, H. Planning as Heuristic Search: New results. In *Procs. of the ECP-99*. Springer (1999).
2. CASTANO, B., AND R-MORENO, M. D. An Artificial Intelligence and RFID System for People Detection and Orientation in Big Surfaces. In *Procs of the 6th International Conference on Computing, Communications and Control Technologies (CCT2008)*. Florida, USA (2008).
3. CESTA, A., ODDI, A., AND SMITH, S. F. An Iterative Sampling Procedure for Resource Constrained Project Scheduling with Time Windows. In *Proceedings of the 16th Int. Joint Conference on Artificial Intelligence (IJCAI-99)* (1999).
4. ENGELS, D. V. RFID: The technical Reality. In *Proceedings of Workshop on Radio Frequency Identification: Applications and Implications for Consumers*. Washington D.C. (2004).
5. GEREVINI, A., AND LONG, D. Plan Constraints and Preferences in PDDL3. The Language of the Fifth International Planning Competition. Tech. Rep. Technical Report, Department of Electronics for Automation, University of Brescia, Italy, 2005.
6. GEREVINI, A., S. A., AND SERINA, I. An Approach to Temporal Planning and Scheduling in Domains with Predicable Exogenous Events. *jair* 25 (2006), 187–213.
7. HALSEY, K., LONG, D., AND FOX, M. CRIKEY - A Planner Looking at the Integration of Scheduling and Planning. In *Procs. of the Workshop on Integration Scheduling Into Planning at 13th International Conference on Automated Planning and Scheduling (ICAPS'03)* (2004), pp. 46–52.
8. HOFFMANN, J., AND NEBEL, B. The FF Planning System: Fast Plan Generation Through Heuristic Search. *Journal of Artificial Intelligence Research* 14 (2001), 253–302.
9. PLAZA, J., R-MORENO, M. D., CASTANO, B., CARBAJO, M., AND MORENO, A. PIPSS: Parallel Integrated Planning and Scheduling System. In *The 27th Annual Workshop of the UK Planning and Scheduling Special Interest Group (PLANSIG05)*. London (UK). (2008).
10. R-MORENO, M. D., CAMACHO, D., AND MORENO, A. HPP: A Heuristic Progressive Planner. In *The 24th Annual Workshop of the UK Planning and Scheduling Special Interest Group (PLANSIG05)*. London (UK). (2005).
11. SABELFELD, K. K. *Monte Carlo Methods in Boundary Value Problems*. Springer-Verlag, 1991.
12. VIDAL, V., AND GEFFNER, H. Branching and Pruning: An Optimal Temporal POCL Planner based on Constraint Programming. *Artificial Intelligence* 3, 170 (2006), 298–335.