

# Modelling Agent Systems Using the Hotel Analogy “Sanitised for your Protection”

Lindsay Marshall and Savas Parastatidis

Department of Computing Science,  
University of Newcastle upon Tyne, Newcastle NE1 7RU, UK  
Lindsay.Marshall@newcastle.ac.uk, Savas.Parastatidis@acm.org

## Abstract

This paper looks at how a particular social analogy (that of the hotel) could be used to help the design of the environment provided by an agent support system. It discusses some of the implementation issues and problems that the use of the analogy exposes

## 1. Introduction

Analogy is a wonderful tool for finding new models and approaches in computing, and it is useful at all levels from the most conceptual down to the practical. Social analogies in particular are revealing when exploring various aspects of the design space of agent systems. Much insight on how agents could and should interact can be gained from this. However, our concern in this paper is not with agents and their interaction, though we will have something to say about this. For more information on agents the reader is referred to (Genesereth and Ketchpel 1994; Franklin and Graesser 1996; Maes 1994; Maes 1995). We are principally concerned with the environment that supports the agents and how the agents interact with that environment. An environment that must be dependable (i.e. secure, reliable, available etc.), both from the point of view of the agents’ owners and that of the service providers on whose computers the agent programs execute.

An early experiment lead to the development of the *Iris agent execution environment* (Parastatidis 1996) which provided an *office-like* model for the execution of agents. Software components such as *secretaries, receptionists, managers, security advisors, messengers* constituted the building blocks, or the *personnel*, for *branches* and *agencies*. The collection of all of the branch and agency offices formed the Iris agent execution environment. The two different types of offices provided distinct services to the visiting agents. The organisation of the personnel in each office resembled the organisation of a human office system where each member of staff has a specified range of duties.

Evaluation of the Iris agent execution environment lead us to consider other support system analogies and having looked at various options (e.g., libraries, schools, public transport) we decided that one of the most potent is that of the *hotel*. It is this analogy that we will explore in the rest of this paper. Anyone who has ever stayed in a hotel should grasp the reasoning behind our suggestions without difficulty, and be able to see many other extensions that we have missed or which we do not have the space to cover. We see the hotel analogy as a good design model for future agent-based software architecture systems and we show how it can highlight potential areas of difficulty that implementors must consider when working. Please note that much of what we say may seem blindingly obvious. We would (in most cases) maintain that if something is obvious, it is because of the power of the analogy.

The rest of this paper is organised as follows. Section 2 introduces the notion of the hotel while Sections 3 to 8 discuss the design analogies with common actions in hotels. Additionally, we look at potential problems with human guests in hotels that have analogies in the agent world. Finally, Section 9 presents our conclusions.

## 2. The Hotel

Before we proceed in describing the hotel analogy for an agent system, we first need to define what an agent is. For the purposes of this paper, we define an agent as *a program that relocates itself from host to host, carrying out computations at each place it visits*. The exact nature of these computations is not important, but clearly it does provide the reason why particular hosts are

chosen. We will see later how the choice of particular hosts may be influenced.

In our analogy, we regard each host as a hotel in which the agent acquires a *room* for a specified, but possibly extensible, length of time. The room provides the controlled environment in which the agent can execute its tasks. The environment consists of a predefined collection of resources that the agent requests during the check-in process (Section 3). Agents may call for additional resources through the hotel's *room service*. Of course, additional resources are charged to the agent's room. Room service becomes the medium for the interaction between agents and hotels (more on room services in Section 4).

The behaviour of an agent resembles that of a traveller. The agent moves from one hotel to the other acquiring rooms and consuming the resources provided. Also like a traveller, an agent may bring with it *baggage* that provides additional materials that the agent needs to work, but that are not provided by the hotel – when staying in a hotel you usually take your toothbrush, but you will probably not take towels as they will be provided. This baggage is the private property of the agent and as such must be maintained securely and safely.

### 3. Check-in

When an agent arrives at a host where it wishes to operate it must first check-in. (Clearly the check-in process could be null and thus our hotel simply the equivalent of a crash pad, but this is definitely a degenerate case and we shall not consider it further). The queue of agents is processed in order of arrival unless the agent holds VIP status and therefore is subject to special treatment. When the agent's turn arrives it must identify itself and arrange for payment. As always the issue of identity is interesting. Some hotels may operate like a members only club and in these establishing the identity of an agent, and therefore its status as a member or member's guest, is vital. But in other places, it turns out that identity is less important than it might at first seem, as payment is the main issue – after all, you can check-in to most hotels using a false identity as long as you can pay! And then of course there are the sorts of hotels where they are most definitely not interested in your identity at all...

The simplest case is where the agent has a pre-booked room. Having identified itself and thus been associated with the booking, payment must be arranged. It may be that the booking includes billing information already: "Charge company X." However, the agent might have to proffer some kind of e-cash or other credit token which the host

will attempt to authenticate. A suitably intelligent agent may also wish to haggle in order to negotiate a better price ("I'm a member of the AAA1"). The booking will have pre-specified the type of *room* required by the agent, but at this time another type or additional services can be requested ("Can I have a Guardian in the morning please?"). A host can provide support for different kinds of agent through *themed* rooms, thus it could have Java rooms or tcl rooms depending on the implementation base of the agent.

Note that so far this whole process is similar to what went on in batch processing systems where jobs were submitted on cards to the system; JCL cards carried information about the resources needed by the program that followed and the operating system used this information to schedule and control the program.

If no booking has been made, the agent may be turned away because of a lack of rooms or because the hotel will only deal with pre-booked agents. On the other hand, if the agent appears able to make payment for the length of stay requested then the hotel may allow it to remain.

Once the hotel is satisfied that any resources used by the agent can be paid for, and that a suitable room is available, a room will be allocated and the key given to the agent. The room is vital as it provides the agent with the support facilities that it needs to carry out its operations. A hotel may provide many kinds of rooms with prices to match, or, like a Tokyo capsule hotel, one kind of room at a flat rate. It all depends on the service that hosts wish to offer. Clients that pay more will (usually) get better facilities. For instance, a host may provide a stock information service to rooms and will delay providing this information to the occupant by an amount dependant on how much was paid.

If the agent has visited the hotel before then the hotel may know of any special requirements it has and provide them automatically. A hotel may, for example, maintain in its customer database special requests for room services, preferred types of rooms, usual method of payment, etc. The hotel may also keep track of its regular customers and provide them with discounted rates and extra services. Hotels may be parts of *chains* and information about regular users may be centralised and available to all members of the chain.

### 4. Room Service

As indicated above, the level of *room service* an agent gets depends on the amount paid. But what kinds of services might be available? Agents do

not have the same needs as we do! The most obvious service is on a par with a hotel room having a bed—access to CPU power. The agent is there to compute and the amount of CPU time it gets in a given elapsed time will relate to the kind of room it gets allocated. There is no reason why agents should travel alone—you can have the equivalent of double and family rooms for multi-process agents.

Most hotel rooms provide access to information and ours are no different. There will be some free services (for instance date and time), some standard chargeable services (sending and receiving messages over the network) and the equivalent of *pay per view* movies or the *mini-bar* where special access to information is given and a premium charge levied. (Such as the stock market information mentioned above) Once again the level of access and charge rate will depend on the kind of room requested.

Agents can expect any servicing of the room to be invisible and that all their transactions with the system and the outside world are secure, confidential and private. The host should provide facilities whereby an agent can be notified of and receive incoming messages which are addressed to it care of the hotel – note that some of these may come before the agent has arrived or after it has left, these cases must be handled properly with storage, forwarding or just by simple return.

## 5. The Stay

During its stay at a hotel, an agent may only use the resources available in its room. It is not allowed to access any of the resources in other rooms (the privacy of other agents would be compromised otherwise). However, an agent may *invite* other agents to its room. The invited agents may provide additional functionality or just use the room's available resources to communicate and exchange information with each other. In any case, the hotel is not responsible for the invited agents. The hosting agent will have to pay for any additional resources required and also manage the available resources and communication.

A host could also provide for *conference*-style meetings of agents. There may be situations where many agents need to participate in a common task or make information available to each other. For an additional fee the hotel may provide a special room with its own facilities where the agents can meet to work together.

Real travellers rarely stay in their hotel all the time, unless it is a resort hotel or a conference centre, and it is not unreasonable to think about an agent using a host as a local base for information collecting which it then either processes in its

room or takes with it in its baggage on leaving. There may be sound reasons, for instance bandwidth or security considerations, why information may be best accessed using the hotel as a base rather than doing it more remotely.

When dealing with truly social agents that independently interact with others, hotels almost always have bars and restaurants where people meet, and our environment could provide similar facilities for agents where they could interact outside the confines of their room. Quite what this implies depends entirely on the nature of the interactions that the agents themselves can support.

## 6. Support Issues

Hotels are never without troubles from their residents. We expect that malicious or badly behaved agents will exist in our hotel-based agent execution environment and they must be dealt with appropriately. Equally guests often have trouble with their hotel and mechanisms for dealing with this must also be in place. Careful consideration of the workings of a real hotel can reveal many areas where things can go wrong in the hotel/guest interface and the guest/guest interface and we have identified a few of them in this section.

### 6.1 The 007 Problem

Agents may attempt to use their rooms as the basis for spying on other agents in the same hotel. They may try to monitor their activities, the information they hold, the resources they are using. Hotels should not allow this to happen – as we noted above all transactions should be secure, private and confidential. This means that the system must be designed to eliminate as far as possible loopholes and covert channels – there should be no electronic equivalent of bribing the maid to get a key to another room, there should be no connecting doors unless explicitly requested.

### 6.2 Rock Star Problem

We expect that there will be agents that will try to trash the rooms they book and maybe the whole hotel. This will manifest itself in the form of excess resource consumption and means that scheduling and monitoring systems must be carefully designed to prevent this happening, or else to ensure that the host charges the culprits appropriately for what they consume and that other affected guests get some kind of compensation. This kind of problem suggests that an interesting area for exploration is the provision of insurance to services that support agents. After

all, real hotels are all insured against a wide variety of risks so why not a virtual hotel? We are not aware of any company that currently provides this kind of protective insurance to companies.

### 6.3 Barton Fink Problem

What happens if the agent in the room next door turns out not to be as nice as they seem? What does an agent do if it finds something not quite right about its environment? Chains of responsibility must be clearly laid out so that such situations can be investigated and any problems contained. In a hotel environment, there is a web of levels of trust that covers every part of the system. Guests assume that the hotel staff are not stealing from their rooms or looking at their private material. They are probably less trusting of their fellow guests, though they may become more so if they meet. The hotel probably trusts no one.

### 6.4 Garbologist Problem

Another approach to maliciously acquiring information from agents is to examine their execution environment after they have finished. There may be malicious agents that manage to book the same room in a hotel as a target agent that has just checked-out. By examining the room the previous agents left they may manage to acquire important information (e.g., examining the registers of a CPU or reading the memory associated with a particular room). The hotel must activate a *cleaning service* immediately after an agent checks-out that removes any *garbage* an agent leaves behind. Of course the agent may not trust the hotel to carry out this task effectively enough and may itself try to ensure that no traces remain.

### 6.5 Hotel California Problem

What happens if an agent checks out but cannot leave? That is, the agent's ability to move around is compromised by the host. In this state the agent will appear to its owner as having vanished as it cannot communicate, after all it has checked out so has no access to host facilities. If the host is malicious it can answer queries about the agent by saying that it has moved on even though it has not. Some agent programs may be valuable and thus be subject to theft or ransom (codenapping).

## 7. Check-out

When an agent leaves a hotel it should check out. This provides a point where actual resource usage over and above that agreed at check-in can be identified and charges assessed. The agent must negotiate payment for these extras. Equally, it

allows the agent to verify that it (or its initiator) is being charged only for those services that it has used and no others. If it appears that resources that should have been provided have not been then a refund should be asked for at this point.

What happens if an agent has used services that it cannot afford? In a real world, commercial system this could be a real problem – after all a program cannot wash the dishes to pay for a meal! Once again insurance would help, but we would foresee that this would be a stimulus to the increasing provision of member only services where, as we indicated above, the identity of an agent, and thus of its originator, is clearly established at check-in. Knowing this may (though not always) allow recourse to conventional legal means of redress.

Once checked out the agent move to another hotel to carry out more work and if the current hotel is part of a *chain*, it may be possible to have forward bookings arranged by the receptionist instead of the agent having to do this itself. This may potentially give it more privileged treatment at the other hotel – the regular customer analogy.

## 8. Choosing Hotels

Based on the services and the resources the hotels provide, agents can choose which one they want to visit. Thus there will be five star hotels which are expensive but provide the equivalent of unlimited luxury – lots of storage and CPU power, or perhaps some special purpose hardware (though seems more the province of theme hotels). Equally there will be the equivalent of commercial hotels and theatrical boarding houses where agents that travel regular routes can stay economically. The guarantees about security and privacy provided by the hosts will also form an important part of any decision as to which to use.

Clearly the existence of chains where you can expect a known level of service at each member host will be useful when planning the route an agent needs to take to achieve its ends. The chains themselves need not be monolithic organisations, they could be fairly loose groupings of hosts that conform to an agreed set of standards – such collections of independent hotels exist in the real world, though they often tend to be in the more expensive sector of the market.

Ultimately, the choice of where a operator decides to send an agent will depend on many factors, most important being the function that the agent has to perform – there may be only one place that provides the required facilities! A truly independent agent would have to written so that it could evaluate which hotel would be best for it, a challenge that many humans find hard and perhaps another good reason for creating chains of hosts.

## 9. Conclusions

When designing and implementing a new system it is always useful to have an existing model which can provide insight into the kind of problems that one might encounter. Clearly it is essential that an appropriate model is chosen, and one that is sufficiently rich that it will allow the consideration of a wide variety of possibilities. As we stated above, we believe that the hotel analogy provides just such a model for developers of agent systems. Our brief tour of how guests use hotels, how hotels service guests and how this could be applied to agent systems shows, we believe, that the analogy is as powerful as we claimed it to be. We have only really touched on each topic and there is considerable room for the expansion and development of the idea. Everyone's experience of hotels is different and will illuminate different aspects of what is needed from a support system.

## Acknowledgements

Our thanks go to Professor Santosh Shrivastava who first mooted the idea of using the hotel analogy.

## References

- Franklin, S. and Graesser, A. (1996). "Is it an Agent, or just a Program? A Taxonomy for Autonomous Agents." In *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*. Springer-Verlag.
- Genesereth, M. and Ketchpel, S. (1994). "Software Agents." *Communications of the ACM*, 37(7): pp.48-53.
- Maes, P. (1994). "Modelling Adaptive Autonomous Agents." *Artificial Life Journal*, 1(1 & 2).
- Maes, P. (1995). *Designing Autonomous Agents*. MIT Press, Cambridge.
- Parastatidis, S. (1996). *The IRIS agent execution environment*. MSc. Thesis, Dep. of Computing Science, University of Newcastle upon Tyne.