Modelling Interactive Experience, Function and Performance in Ubiquitous Systems

Michael D. Harrison[a][1]   Mieke Massink[b]

[a]*School of Computing Science, Newcastle University, U.K.*

[b]*Istituto di Scienza e Tecnologie dell'Informazione 'A. Faedo', CNR*
Area della Ricerca, Via G. Moruzzi 1, Pisa, Italy

Abstract

The cost of deploying a ubiquitous system to enhance a physical environment is likely to be considerable. The success of its deployment is highly dependent on its context: the physical environment and the activities that are to be carried out within it. This paper provides an initial exploration of whether stochastic process algebras (in particular PEPA with a Fluid Flow semantics) might be used to explore consequences before deployment. The focus of the exploration is to aid understanding of how a proposed system supports users within the environment. The challenge is to provide notations and techniques that will enable the analysis of potentially complex systems.

Keywords: interactive systems, model checking, stochastic modelling, ubiquitous computing, human computer interaction

1 Introduction

This paper takes as its starting point the notion that ubiquitous systems can be designed deliberately to facilitate, or to modify, collective behaviour of people within physical environments. It therefore presumes that designers will require techniques to enable them to predict the impact of the design on those human interactions within the system that make up this behaviour. In Human Computer Interaction the traditional focus for interactive systems analysis techniques has been the user and the device. Typical among these techniques is *cognitive walkthrough* ([7] provides a description of it and other techniques) which assesses the extent to which an interactive device supports typical user behaviours: their goals, their plans to achieve goals, the availability of actions and the clarity of action effects. This paper briefly explores the broader range of properties that may be required to make a

---

[1] Email: michael.harrison@ncl.ac.uk

ubiquitous system effective from a user point of view and the role that stochastic modelling might play in this respect.

The increased availability and resilience of technologies that make up ubiquitous systems is providing an opportunity to create technologically enhanced physical environments that support a range of activities. Potential examples include hospital out-patients, where the role of the system would be to guide and inform the patient at each stage of a consultation, or an airport where the airlines provide information to passengers about flights, the airport authorities provide services related to the building and security, and the retailers about food, duty free etc.

Many factors are important if such environments are to be successful and appropriate. Some of these factors may be identified and built into the design using conventional techniques for interaction design, for example techniques related to the usability of an individual device or display. Such techniques involve the analysis of typical behaviours represented through the use of scenarios or tasks [21]. Ubiquitous environments however provide additional challenges for usability analysis. They use sensors to identify individuals (using sensing tags or mobile Bluetooth enabled devices) and use their context (for example location, destination, prior behaviour, task, intention) to tailor the information that is provided to individual devices or to public displays. They support activities implicitly through inference of an individual's changes of context, anticipating their needs. Features such as these have the effect of immersing users within the system, making it possible to create or design an *experience* for users within a technology enhanced environment.

The implicit actions that a system might take will depend on a number of factors, for example whether the individual has a mobile device, what its characteristics are, and who else is in the space. The MATCH system [17], for example, is designed to provide assisted living for people with special needs. When a potentially adverse event is detected a number of actions are taken by the system depending on context. If the carer is in the same room as the individual then the system does nothing. If the carer is in the same house but not in the room, then the carer is alerted either by sound or by a message on a mobile device depending on the ambient conditions in the room where they are located. If the carer is not in the same house then the surveillance centre for the sheltered housing is informed.

Traditional forms of usability analysis are not sufficient to analyse these systems in terms of the experience that they create. Complementary techniques are required that focus on issues such as enjoyment, awareness of safety or security, or providing a sense of place in the physical environment. These complementary techniques are already a concern in the wider design community as discussed in [1]. The question that motivates this paper is how to understand and design for experience, and what role, if any, modelling techniques in general and stochastic modelling techniques in particular would have in that analysis.

There are many features of designs of environments that may have an impact on the experience (enjoyment, frustration etc.) of users of the environment. Consider the following examples:

Visitors to an office building, or to a stadium to watch a football game:

will need information about how to get to a specific office, or seat within the stadium. Timely and relevant information tailored to individual needs will enable visitors to reach their destinations without confusion or frustration.

People evacuating an office building, or stadium: will need clear and calm instructions at each stage of their exit to ensure they take optimal routes that offer least congestion.

Airline passengers: will require notification about which queue to join for check-in, baggage screening, passport control etc. This information, provided individually at each stage about what is expected of them, could reduce waiting, and improve a sense of the airport as a place rather than a forbidding space.

Hospital out-patients: will require directions depending on the nature of their appointment and the stage they have reached, to have a blood test, or an x-ray, or a scan, or to see the doctor, or attend a wellness check.

Newly registering college students: will need directing to get a library card, pay fees, see their tutors, as part of the registration process at the start of their courses

Experience in these systems might also be improved through a number of techniques to deal with idle time, for example in a stadium or airport visitors could be advised about retail or entertainment opportunities to fill idle time before the event / boarding. Hospital out-patients might be informed about the availability of routine checks if they are waiting and eligible and there is spare capacity to provide these checks. A model that only aims to capture the functional characteristics of these ubiquitous systems may not be sufficient to address the experience requirements discussed. The texture and physical characteristics of the enhanced environment, along with the pervasive ambient background to the system, is crucial to the experience of being within it. If these requirements are to be expressed of a new design then an important question is how these requirements can be expressed for implementers so they can be discharged in the design. The literature on experience requirements (see for example, [1]) suggests a number of stages in establishing and expressing requirements. The first stage is how to get information about what would create or destroy a desired experience. This can be achieved by obtaining from typical users (or particular types of user) narratives or scenarios either where the existing system has created a negative experience or where something positive or desirable has happened. Scenarios might be accounts of recent poor experience, for example concerning a delayed flight where little information was provided about the extent of the delay, or the availability of alternative flights, long queues, no account taken of the passenger's connection time at the next airport. The scenario can be a generator for the new design, helping to avoid the pitfalls of the described scenario. An alternative, complementary approach is to ask individuals to record their experiences (using sound recordings or producing photographs, with comments to explain the significance of the recorded information) as they carry out the activity in the existing system.

The second stage in this process is to use the information productively in design.

The narrative can be used to explore a new design alternative. Revised narratives can be constructed that incorporate a description of how the envisaged design may be used. These narratives can be used to explore with potential users what the new design would be like, enabling them to visualise its impact. This visualisation is similar to the sort of process that is commonplace within the broader design community, where, for example, clay models represent a proposal and are used or visualised by subjects in their day to day activities [1]. Information derived from the first stage of the process may also include an understanding of the desirable properties of the design. These properties might relate to the siting of information displays in relation to the seating in the room, it may relate to what information is required at what stages of the process, it may relate to the acceptability or otherwise of delay, the necessity to have regular updates of the information. Here functional models would enable the description of these relationships, expressed as properties, providing a basis for analysis. The results of the model could be counterexamples where properties break. These counterexamples could be animated and thereby visualised to generate further scenarios for consideration relating to the proposed new design.

The paper focusses on challenges associated with developing functional models of a class of ubiquitous systems whose design may benefit from qualitative and quantitative reasoning. The paper first describes a class of systems that may be susceptible to this analysis (Section 3). It demonstrates the class through two examples. While a variety of properties of such systems may relate to the user's experience of these environments, the paper focuses particularly on challenges for quantitative modellers (Section 4).

2   Related work

The approach taken in the paper is concerned with the use of formal modelling and associated analysis techniques of interactive aspects of ubiquitous systems. There is already a substantial literature on the role of formal methods in the design and evaluation of interactive systems. Much of the focus of current formal analysis has been based on model checking or other algorithmic techniques, for example graph analysis. One motivation for the analysis has been to find instances of behaviour that could be problematic that are then analysed by human factors and domain experts. The domain experts generate a richer description of the working of the system based on the results of the analysis, and the human factors experts explore scenarios based on traces. Typical properties of interest include predictability, visibility, the ability to recover to a previous state and clarity of mode structure. Models focus on key features of interaction in the system, using notations such as modal action logic (MAL) [2], statecharts [6] or matrices [9]. Models and analyses for Flight Management Systems [23,2], in-car control systems [3] and medical instruments [24] have been carried out, in each case predicting unforeseen consequences of the design.

Further notations are relevant to ubiquitous systems to model the whole environment. This means capturing:

- the spatial layout of the environment, suggests defining the logical properties of mobility and space (for example, KLAIM and π-calculus [5,19] and Ambient Calculus [4])

- the properties of visitors, for example assuming goals for the visitor or defining error behaviour, relevant research here explores models of human cognitive processes, see for example [22]

- context that can be used to make implicit action.

Relevant research is also concerned with the formal analysis of protocols relevant to ubiquitous systems, see for example [8] in relation to publish-subscribe and [16] in relation to gossip protocols.

The characteristics of human interactive behaviour in ubiquitous systems need to be expressed functionally but also in a form that makes it possible to explore the stochastic behaviours of multiple occupants of the space.

3   Examples: guidance and out-patients

A class of ubiquitous systems forms the basis for the proposed quantitative analysis of user related properties. The systems in this class aim to enhance buildings by providing information for the guidance of those within the environment and are applicable to a range of domains. Two examples are used to illustrate commonalities: a guidance system and a system to assist out-patients in a hospital.

3.1   The guidance system

The first system guides visitors who are unfamiliar with an office building to a particular location in the building. The only *deliberate* action required by a visitor is to determine their destination somehow. This could be done by means of a visitor's badge or the visitor's diary on their phone for example. The system uses a set of situated displays attached to walls and doors in the building. It is inspired by an existing system installed at Lancaster University [14]. Directions are displayed on situated displays (see Figure 1 for an example of the existing display) which change depending on where the people who are in the vicinity of the display want to go. Depending on who is present, the display indicates a direction or directions that are relevant to them. Such a system could be installed in a museum, concert hall or stadium, where people arrive in great numbers and are generally unfamiliar with the location of their seats or some other desirable location.

Several people may be in a particular space at the same time. For this reason more than one direction is displayed along with its associated destination, so that the visitor can choose the one that is relevant to them. Since collective behaviours are of concern it is assumed that a space has a specific capacity. Having arrived at a space where there is a display there are four possibilities.

(i) The direction relating to the person's destination is already visible on the display (it is assumed that the person will read the display and see the destination
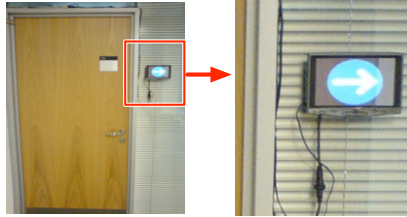
Fig. 1. The situated display

displayed as appropriate).

(ii) The relevant direction is not currently displayed but there is an empty slot on the public display. In this case the relevant direction with destination specified is downloaded automatically (as far as the visitor's perception of the display is concerned this case will appear to be identical to (i)).

(iii) The relevant information is not displayed and there are no empty slots. If the space is not full of people (say there are less than 25 people already waiting) the person will wait, regularly reading the display, until the display is eventually updated to show the required destination.

(iv) When there are no empty slots, the destination is not represented, and the capacity of the space has been exceeded, the person makes a random move to another space.

3.2 The out-patient system

The second example is designed to assist out-patients at a hospital. A patient arrives with an appointment. The aim of the ubiquitous system is to provide information and guidance to patients so that they are (1) kept up to date with information about waiting times (2) guided to the next stage of their patient journey when it is necessary for them to move on. Patients may have mobile devices, in which case the information that is relevant to them is sent direct to their device, or do not have appropriate mobile devices in which case they are issued with a sensor/visitor card which can be used by the environment to recognise what they must do next. In the latter case public displays are updated with the number of the patient and information about wait times or guidance about where to go next. A simple system might recognise four types of patient:

(i) patients who need to see the doctor (D) only

(ii) patients who need an X-ray (X) then to see the doctor (D)

(iii) patients who need a blood-test (B) and to see the doctor (D)

(iv) patients who need to have a blood test (B) and an X-ray (X) before seeing the doctor (D).

In addition a wellness check is offered to certain categories of patient if it is possible to complete the check before the next stage of their process. Relevant information will either be sent to the individual's mobile device or to a public display in the space
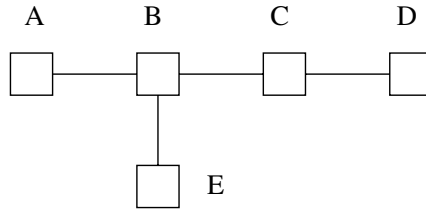
Fig. 2. Layout of the building

that they occupy if they do not have the mobile device. The system will reschedule patients of type (iv) so that the most available of (B) and (X) is scheduled. If there is time then a wellness check is offered (W).

3.3 Commonalities between designs

Both examples have common features. They involve individuals reaching destinations. In one case the route to the destination is determined by the system once the destination of the visitor is recognised, in the other it depends on patient type, all patients aim to reach the same destination. Hence both fit into a general class of systems which push information depending on context. In each case people respond to the information by taking action that changes their context in some way.

4 Modelling the Guidance System

The first example will be used to explore what might be appropriate properties of ubiquitous systems. As has been discussed already a number of properties concerning a user's experience depend on characteristics of the environment including its texture, knowing what is happening next, being aware of where things are and so on. Often experience is related to the rate of flow and relevance of the information in the system [18] and what happens when there are misunderstandings or mistakes. An important characteristic of user experience is that it is determined by the circumstances of the context.

4.1 Properties

In the case of the guidance system the only *deliberate* action taken by a visitor (it could be a patient) is to purchase the electronic ticket some time before the event, or to obtain the visitor pass at reception, or to bring a tagged visitor's card, or to bring a Bluetooth enabled mobile phone. Otherwise all interactions with the system occur implicitly as visitors change their context, for example by moving from room to room or corridor to corridor. Experience of the environment depends on this information. The visitor or patient is involved in a journey, either a patient journey from arrival at out-patients to departure after consultation, or a visitor journey from reception to the destination in the office building.

A number of factors could have an impact on the person, or people, in the environment. Their relevance depends on the particular context and would be assessed through some form of user evaluation. These factors could include:

  (i) the visibility and interpretation of display directions

 (ii) the continuing visibility of directions wherever the user is

(iii) a sense of progress towards the destination

(iv) the ability to remember the route having completed it once

 (v) a broader sense of the building and the facilities it offers

(vi) how long to wait before the display is relevant to them (either in terms of time or number of refreshes of the display)

(vii) guaranteed time to arrival

(viii) impact if many users need to recover

(ix) a sense of congestion, that there are too many people in the surrounding space.

All these factors are likely to affect the level of satisfaction or anxiety amongst users of these environments. Their experience however may also be dependent on external factors and the expectations users bring with them. Other processes that cannot be modelled within the design are implicated. However it is only by recognising and integrating factors that relate the system design to these external factors that progress can be made. While the factors at the top of the list are more familiar in the context of usability, those towards the end of the list are more novel in content from a human-computer interaction perspective. These properties have the following characteristics:

- They are about providing relevant information when and where it is needed as well as providing a sense of progress towards the destination.

- They concern how specific routes and guidance can be used to reduce the likelihood of bottlenecks.

Implicit interaction in the guidance system may be modelled either by bringing into focus an individual or a small group of individuals within the system, or by focussing on collective behaviour while abstracting from individual behaviour. A model is only valuable in design if it has formative value. It should allow the exploration of the impact of design options and enable a comparison between options. If this can be done effectively before the design is deployed then downstream costs are likely to be reduced. This analysis, while technical, should be presented in a form that is meaningful to those who are concerned with human factors aspects and will visualise the design represented by the model.

5   Exploring a subset of properties

The advantage of an approach using a stochastic process algebra such as PEPA [12] is that it is possible to combine a functional model that captures key design assumptions with techniques that enable analysis of stochastic behaviour. This eases the constructive use of the results in further design iterations. PEPA will not be described in detail. The proposed PEPA specification is based on a qualitative model described in PROMELA [13] that captured more of the functional details of

the system [10]. Simplifications in comparison with the qualitative model include a focus on spaces rather than locations. For example, the possibility that a visitor might be in a location where there is no sensor or display is ignored. The model "wires in" the directions within the location based processes rather than being requested from a centralised process.

The specification is modular and is driven by a number of parameters. A purpose-built program generates models based on the organisation of the space, the number of slots, places and individuals within the system. The input to the program describes for each location (1) the number of places that it supports and the number of slots the display contains; (2) for each destination the visitor's next location. The file also describes the number of visitors that start in a specific location requiring a particular destination. This information enables the generation of an appropriate PEPA specification and it has a form that also makes it amenable to Ordinary Differential Equation (ODE) analysis [11] building on the ODE-based formal semantics for PEPA and related analysis tools [25]. The illustration in the paper is restricted to five locations for the purposes of concise presentation. Two of these locations are final destinations. The locations are called a, b, c, d and e, and are connected in the following way: (a, b), (b, c), (c, d) and (b, e) as shown in Figure 2. Larger configurations have also been analysed and the approach has been shown to be feasible for buildings with a realistic number of locations and visitors.

This quantitative model otherwise reflects the same structure and character as the qualitative model described in [10]. Each location has a number of places and each display a number of slots which can be consulted by visitors that are present in the location. The display is combined with a sensor, detecting where a visitor wishes to go and displaying relevant information. Three kinds of processes model the behaviour of a visitor, display and place.

The PEPA model consists of processes modelling the behaviour of visitors, arbitrators, slot managers, slots and places. Slots and places are instantiated for each particular location. The display consists of a number of slots. The slot manager and the arbitrator, in each location, ensure that requested information is displayed and that no two slots show the same information. Groups of visitors are defined with the same starting location and final destination. For example, a visitor starting in location a and heading for location d first tries to get a place in location a where it is possible to see the display ($lasd$ is the action of trying to acquire a place in location a). Once a place has been acquired the visitor engages in action $laee$ to find out where to go next. The request is engaged as soon as there is an available display slot that displays the information. When the information is displayed the visitor releases the place in location a (action $lasu$) and receives the information (i.e. any of the matching destinations in the process $VisEdRec$). The visitor then proceeds to the indicated next location (e.g. $VisEdtoLb$ means that the incomer with final destination d now first needs to proceed to location b). The arrival at the final destination is modelled by the process that remains in the state $VisEdArrived$ forever.

The specification of the visitor process in PEPA is of the following form (see [10]

$$
\begin{aligned}
VisEytoLx &= (lxsd, s).(lxey, a).(lxsu, s).VisEyRecx \\
VisEytoLy &= (lesy, s).(lyey, a).(lysu, s).VisEyArrived \\
VisEyRecx &= (eexle, r).VisEytoLe+ \\
&\quad (eexla, r).VisEytoLa+ \\
&\quad (eexlb, r).VisEytoLb+ \\
&\quad (eexlc, r).VisEytoLc+ \\
&\quad (eexld, r).VisEytoLd \\
VisEyArrived &= (nop, a).VisEyArrived
\end{aligned}
$$

Fig. 3. The general form of visitor in PEPA

for full specifications). In the actual specification these processes are repeated for the combinations of locations. The general form for visitor $VisEytoLx$ currently at location x, wishing to reach destination y, is defined in Figure 3.

The model of a visitor has three rate parameters, modelling the average time needed to perform the related activity. The average duration of activities is defined by their rates and is assumed to be measured in minutes. So, for instance, letting $s = 10$ implies that the average time a visitor needs for sitting down or standing up is 6 sec. (i.e. 0.1 min.). Rate $a = 2$ models the average time a visitor needs to make a request equal to 30 seconds. The rate $r = 1$ models the average time to receive the requested information equal to 1 minute. It is further assumed that visitors are arriving over a certain period of time and heading for different destinations. This is modelled in the following way:

$$
Vis = (nop, v0).VisEetoLa + (nop, v1).VisEdtoLa
$$

The action nop stands for a dummy no-operation action. The rates $v0$ and $v1$ can be used to tune the rate of generation of the two types of visitors (i.e. those heading for e and those heading to d) and their relative number. For example, in the case of 60 visitors in total, the number of visitors heading for location e, i.e. the number of $VisEetoLa$ processes, will be $v0/(v0 + v1) \times 60$ while the number of $VisEdtoLa$ processes amounts to $v1/(v0 + v1) \times 60$. It should be noted that this models an arrival pattern of visitors in which most visitors arrive 'early' and, while time passes, there are a diminishing number of visitors arriving.

Each location uses an arbitrator process that displays directions to destinations depending on requests from visitors. The arbitrator ensures that a slot in a location does not show information that is already displayed by another slot in the same location. Process $ArbLaEe$ starts with an empty display and waits for a request for information from a visitor. If such a request arrives ($laee$) it sets up a slot to display the required information by sending a request to the Slot Manager process ($slaee$). Further requests for the same information are then no longer handled by the arbitrator but directly by the relevant slot. Requests for routing information

$$SlotLa \qquad = (sslaee, 100.0).(eealb, r).SlotLadispEe+$$

$$(sslaed, 100.0).(edalb, r).SlotLadispEd$$

$$SlotLadispEd = (laed, a).(edalb, rr).SlotLadispEd + (freelad, f).SlotLa$$

$$SlotLadispEe = (laee, a).(eealb, rr).SlotLadispEe + (freelae, f).SlotLa$$

$$SlotLe \qquad = (sslee, 100.0).SlotLe + (ssleed, 100.0).(edelb, r).SlotLedispEd$$

$$SlotLedispEd = (leed, a).(edelb, rr).SlotLedispEd + (freeled, f).SlotLe$$

Fig. 4. A display slot at location a

for different destinations are still handled by the arbitrator which will start up new slots via the Slot Manager as long as there are empty slots available. Slots become free and the arbitrator is informed via the Slot manager ($sfreelae$) and returns to its initial state. The activities internal to arbitrators and slots are considered to be relatively fast, which explains the chosen fixed rate 100.

$$ArbLxEy = (lxey, a).(slxey, 100.0).(sfreelxy, 100.0).ArbLxEy$$

If a location is only a final destination (so no visitors for other destinations pass through this location) the arbitrator does not need to set up a display slot for an individual:

$$ArbLeEe = (leee, a).ArbLeEe$$

The Slot Manager ($SmanLxy$) initialises and releases slots. It receives a request from the arbitrator ($slxey$) of the location to set-up a slot that displays routing information to a particular destination. The request is followed by an initialisation message from the Slot Manager to a free slot ($sslxey$). That the slot is now free is notified to the Slot Manager ($freelxy$) and then forwarded to the arbitrator of the location that keeps track of free slots and the information on display ($sfreelxy$). The process for a slot manager at location x dealing with requests for destination y is:

$$SmanLxy = (slxey, 100.0).(sslxey, 100.0).(freelxy, f).(sfreelxy, 100.0).SmanLxy$$

A display slot for location a (Figure 4) can be set to display direction information for a particular destination after a message from the Slot Manager, for destination d ($sslaed$) and destination e ($sslaee$). The process handles requests for destination d after the slot has been set up for that destination. When a slot already displays the directions for a particular destination then the response is much faster, given by rate rr. The information can be removed from the display by an autonomous action of the slot called $freelxy$ labelled by the relevant location (x) and destination (y) for which it was displaying information.

Whether it is possible for a visitor to see the display (i.e., a place is free) is modelled by:

$$PlaceFreeLa = (lasd, s).(lasu, s).PlaceFreeLa$$

$$(SmanLxy[1] \parallel Vis[400]) \bowtie L(SlotLx[2] \parallel ArbLxEy[1] \parallel PlaceFreeLx[100])$$

where:

$$L = (Requests \cup Responses \cup ArbReqSlot \cup StartSlot \cup$$

$$ArbFreeSlot \cup ReleaseSlot \cup Sitdown \cup Standup)$$

$$Requests = \{lXeY | X \in \{a, b, c, d, e\}, Y \in \{a, c, e, d\}\}$$

$$Responses = \{eY XlZ | X, Z \in \{a, b, c, d, e\}, Y \in \{a, c, e, d\}, y \neq x\}$$

$$ArbReqSlot = \{slXeY | X \in \{a, b, c, d, e\}, Y \in \{a, c, e, d\}\}$$

$$ArbFreeSlot = \{sfreelXY | X \in \{a, b, c, d, e\}, Y \in \{a, c, e, d\}\}$$

$$StartSlot = \{sslXeY | X \in \{a, b, c, d, e\}, Y \in \{a, c, e, d\}\}$$

$$ReleaseSlot = \{freelXY | X \in \{a, b, c, d, e\}, Y \in \{a, c, e, d\}\}$$

$$Sitdown = \{lXsd | X \in \{a, b, c, d, e\}\}$$

$$Standup = \{lXsu | X \in \{a, b, c, d, e\}\}$$

Fig. 5. The PEPA model of the system

In the example for analysis there are four kinds of visitor. One kind starting in a heading for d, one starting from c heading to e, one starting from a heading to c and the last kind starting from d and heading to a. There are 400 visitors in total, so with the specified rates for $v0$ to $v3$ the groups are composed of respectively 25, 75, 100 and 200.

$$v0 = 0.0625; v1 = 0.1875; v2 = 0.25; v3 = 0.5;$$

$$Vis = (nop, v0).VisEdtoLa + (nop, v1).VisEetoLc +$$

$$(nop, v2).VisEctoLa + (nop, v3).VisEatoLd$$

The overall system is given by the PEPA composition expression in Figure 5. It composes 400 visitors, divided into the four different kinds, and 100 places and 2 slots per location. The processes synchronise over all the actions listed with the cooperation operator. There is one *SlotManager*, modelling the local display, and one *arbiter* per location x for each destination y.

6   Analysing the model

Questions and challenges relate to the sort of analysis that can be done using this model and whether these techniques are relevant to the properties discussed in the earlier section. For this small example the design parameters that are being adjusted include the design of the display, for example how many slots should it contain at the same time. It would be concerned with the size of the display, how many can see it at the same time. It should enable an exploration of the use of different paths and how these paths affect the congestion of the system. This would support the design of

dynamic scheduling algorithms aimed at reducing the congestion and waiting times. Some of these properties may be explored using model checking, while others may be explored through simulation or techniques based on differential equations. The issue here is how can a range of techniques be used to explore these experience requirements of the design. Further work is to be done. The following examples of analysis are preliminary and are included as a challenge rather than a set of results.

6.1 Analysis with PRISM

PRISM [15] was used to verify a number of simple qualitative and quantitative CSL properties. This enabled a credibility check while at the same time exploring the paths to destinations that are assumed by the model to ensure that they are appropriately connected.

For example, the analysis was used to check the probability that different instances of processes arrive at destinations d and e, or that they occupy particular locations. In addition (for the limited number of instances of visitors that were explored in the model checking case) the role of a particular location was explored as a step in multiple paths. The probability of visiting location c which is likely to be true because c is required for both destinations:

```
P=? [ true U (PlaceFreeLc_STATE=PlaceFullLc)]
// Result: 0.996
```

The model was used to determine the probability that a visitor will backtrack or not pass through a necessary location. It was also used to explore the likelihood that two different instances of the visitor would both arrive at destinations e and d within 5 time units:

```
P=? [ true U<=5 (Vis_2_STATE=VisEeArrived & Vis_3_STATE=VisEdArrived)]
// Result: 0.0026523
```

These examples give a very preliminary indication of the role that stochastic model checking can play. However it can be seen that some of the characteristics of the environment relevant to the experience that users could have within it can be explored. This analysis serves a similar role using a more abstract model to that played by the qualitative model checking reported in [10]. However in the present case the same model can be used for verification of collective behaviour.

6.2 Analysis based on stochastic simulation

Simulation and Fluid Flow analysis techniques can also enable constructive reflection on design. These techniques will be illustrated using the building shown in Fig. 2 for 400 visitors divided into 4 groups entering from locations a, c and d and with final destinations a, c, d and e. The model is described in the previous section. Results are shown for stochastic simulation with 10 independent replications and a confidence interval of 0.05. A further aim is to illustrate the correspondence of results between the two techniques and thus the value of using the less computationally intensive Fluid Flow techniques. A summary of values for the parameters

Rate			Meaning
s	$=$	10	On average a visitor takes 6 seconds to sit down on a free seat
a	$=$	1	On average a visitor needs 1 minute to request information
r	$=$	2	On average a slot takes 30 seconds to display the required info
rr	$=$	10	On average a visitor takes 6 seconds to read indications already show
f	$=$	0.2	On average information remains 5 minutes on display

Table 1
Definition of parameter values

of the model are defined in Table 1. Time units are in minutes.

Figure 6(a) shows the number of places occupied in the various locations by the visitors while they move from the locations from which they enter the building to their respective destinations. The first observation is that in location d the maximum number of places is being occupied for a certain amount of time (to be precise, for about 100 minutes approximately as shown by the related curve). In the other locations there are almost always enough free places available for visitors. The correspondence between the simulation results shown in Figure 6(b) and the results obtained via ODE (Fig. 6(a)) suggests that the ODE results are predictive. The results are shown for a time interval ranging from 0 to 300 minutes.

Figure 7(a) shows the time it takes visitors to arrive at their respective destinations. It can be seen that after 50 minutes all visitors with final destination d have arrived. Note that the arrival time depends also on the rate at which these visitors enter the building and not only on the time that visitors spend in each location. The figure shows that the total number of visitors for destination d is 25, that for e is 75, that for c is 100 and that for destination a is 200.

Figure 8(a) and Figure 8(b) show the average number of empty slots on the displays in the various locations. Initially all slots are empty, but they rapidly get occupied when visitors arrive at the locations. When all visitors have arrived at their final destinations and the information on the displays has been removed, all slots return to their empty state. The displays have at most two slots occupied even though there are four different destinations. In location d only at most one slot is occupied because only visitors heading for a single destination pass by. In location e the slots remain free because it is only a final destination in the current model, so no visitors ever need to get information on where to go next.

These preliminary results show some promise. They provide insight into the impact of a ubiquitous system design on the congestion to users in different traffic situations. The design of the system can be assessed in terms of different assumptions about the size of display (how many can see it), the number of slots on the display and the routes that are defined for reaching the different destinations. These variables can be manipulated. This analysis could be used for example as a basis for changing paths dynamically within the ubiquitous system to create a situation
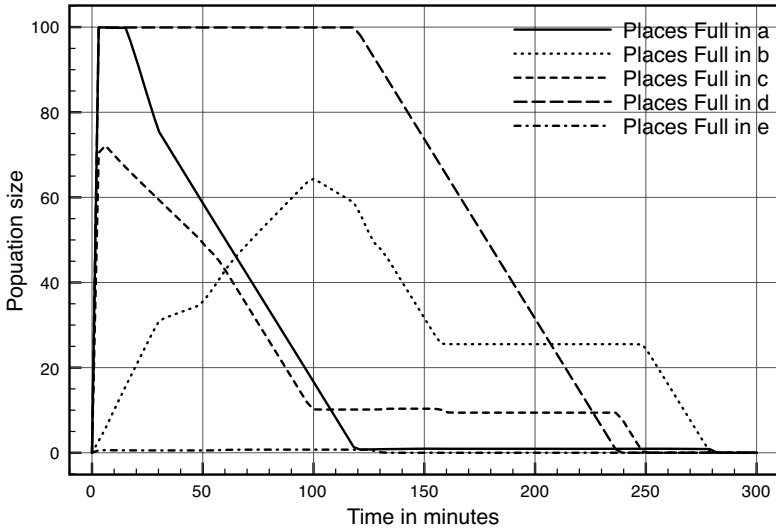
(a) Occupied places over time (ODE approx.).
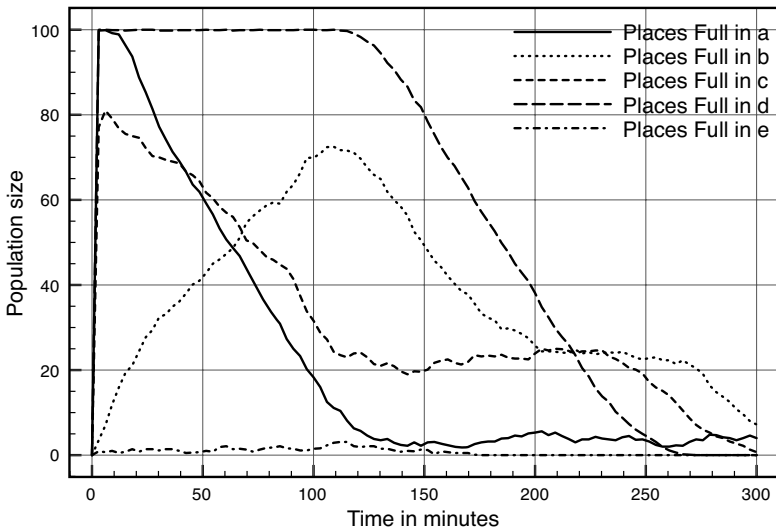
(b) Number of full places/seats over time (stochastic simulation).

Fig. 6. Analysis of places occupied

that will reduce user frustration or anxiety. Combining these results with other functional properties of the system, for example how the displays are sited, when and where the information is sent, typical delays before seeing an individual's destination on the display, will enable a clearer understanding of the design before it is deployed.

The results in this section also show that the ODE technique can deal with a high number of visitors heading to various destinations and starting from various loca-
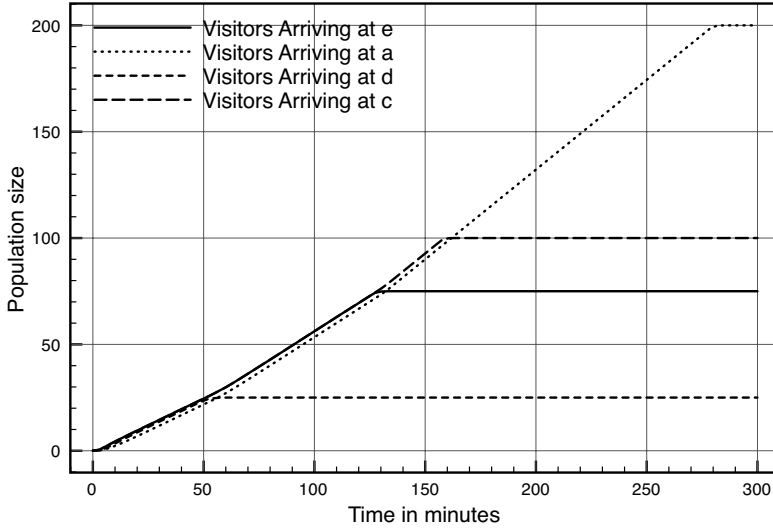
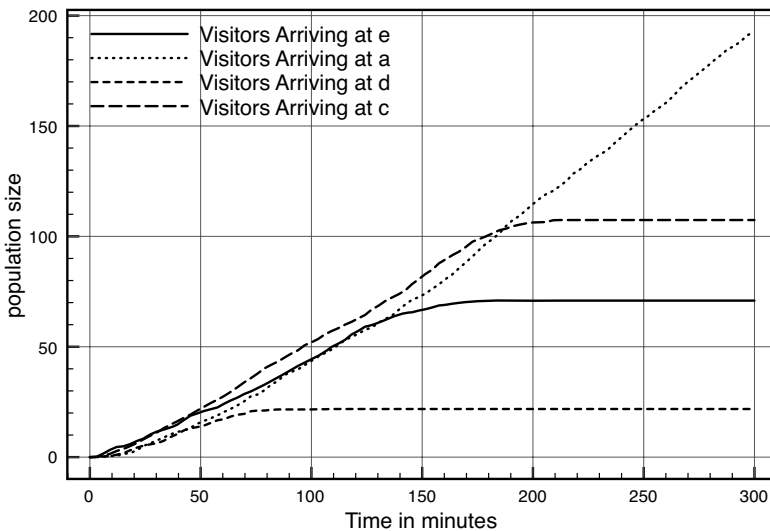(a) Visitors arriving at destinations (ODE approximation).

(b) Number of visitors arriving at a, c, d, and e (stochastic simulation).

Fig. 7. Analysis of visitors arriving

tions. This is because it represents visitor processes, and their states, as continuous quantities instead of as a large set of discrete entities with interleaving behaviour. While simulation can always be applied to the PEPA specifications, it can be prohibitively time-consuming if a high level of accuracy is required. This makes the ODE approach an attractive alternative to simulation to explore the effect on visitor flows for different design options during early design phases. The reliability of the results obtained with ODE can be easily (and often quickly) checked by comparing
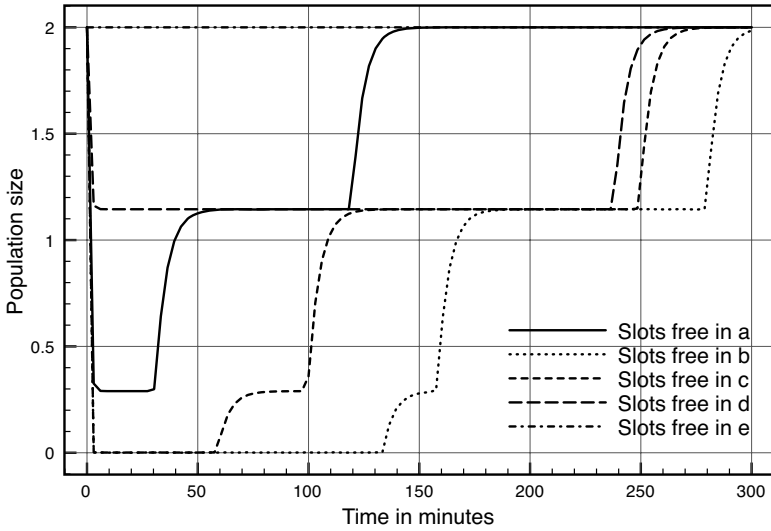
(a) Free slots in display in each location (ODE approx.)
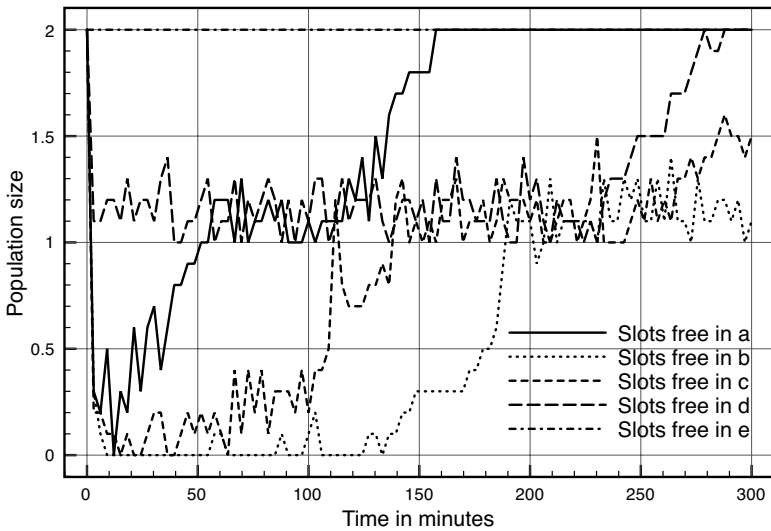
(b) Number of free slots over time (stochastic simulation)

Fig. 8. Analysis of free slots.

the results with a 'limited' simulation where only a few independent executions are considered. If there is a reasonable correspondence, then the ODE results can be considered sufficiently predictive. If there is no reasonable correspondence, then, for that particular analysis, it is always possible to resort to simulation without any change of the model if that is considered worth the time it will cost.

7   Discussion and Conclusions

This paper indicates two directions that are proposed for the future development of human interaction engineering techniques for technologically enhanced environments. The first direction is towards a broader understanding of what usability means in such environments. While more conventional usability analysis is required to assess interaction between device and user, further techniques are required to deal with the implicit interactions that take place in these systems as well as the particular requirements and concerns of crowd-system interaction. By making interaction requirements explicit, modelling can also pave the way to define more clearly what is meant by 'calm computing' [20] in the context of a particular system. The next step is to integrate and use the approach described in this paper in the context of a larger and more complex project. The second direction addresses the problem that it is not usually possible to explore a system in its target environment. It can be difficult to envisage what the impact of a proposed design would be in such an environment. At the same time stakeholders will require strong justification for the introduction of a new system in a commercially or safety sensitive environment (and airport or accident and emergency for example) before such a system can be deployed.

The paper has explored the contribution a PEPA model could make in analysing novel human aspects of smart environments where many people are present. The presence of many users moving at the same time through the same space and interacting with it in an implicit way, and under certain time-constraints, poses special challenges. As has been illustrated an approach based on Ordinary Differential Equations, derived from process algebraic specifications modelling both large groups of users and the environment, promises to provide a complementary quantitative analysis of human aspects of collective behaviour of smart environments that may also be formative in design. Using this technique different assumptions about volumes, rates and design parameters can be compared rapidly with appropriate accuracy.

It remains to be seen whether investment in relatively complex and sophisticated techniques (compared with discount usability engineering techniques often used in the design of interactive systems) will be seen by the designers of technology enhanced physical environments as justifying the effort of using them. Future research is required to reduce these costs, to develop processes, patterns of properties, and generic smart environment models that will make the task of automatically synthesising these models possible and convenient for developers.

Acknowledgement

References

[1] M. Buchenau and J.F. Suri. Experience prototyping. In *Proceedings Designing Interactive Systems (DIS'00)*, pages 424–433, Brooklyn, New York, 2000. ACM Press.

[2] J.C. Campos and M.D. Harrison. Model checking interactor specifications. *Automated Software Engineering*, 8:275–310, 2001.

[3] J.C. Campos and M.D. Harrison. Systematic analysis of control panel interfaces using formal tools. In N. Graham and P. Palanque, editors, *Interactive systems: Design, Specification and Verification, DSVIS'08*, volume 5136 of *Springer Lecture Notes in Computer Science*, pages 72–85. Springer-Verlag, 2008.

[4] L. Cardelli and A.D. Gordon. Mobile ambients. In M. Nivat, editor, *Foundations of Software Science and Computation Structures*, number 1378 in Springer Lecture Notes in Computer Science, pages 140–155. Springer-Verlag, 1998.

[5] R. De, Nicola, G. L. Ferrari, and R. Pugliese. KLAIM: a kernel language for agents interaction and mobility. *IEEE Transactions on Software Engineering*, 24(5):315–330, 1998.

[6] A. Degani. *Taming HAL: designing interfaces beyond 2001*. Palgrave, Macmillan, 2003.

[7] A. Dix, J. Finlay, G. Abowd, and R. Beale. *Human Computer Interaction*. Pearson Education Ltd., 3 edition, 2004.

[8] D. Garlan, S. Khersonsky, and J.S. Kim. Model checking publish-subscribe systems. In *Proceedings of the 10th International SPIN Workshop on Model Checking of Software (SPIN03)*, Portland, Oregon, 2003.

[9] J. Gow and H.W. Thimbleby. MAUI: An interface design tool based on matrix algebra. In R. J. K. Jacob, Q. Limbourg, and J. Vanderdonckt, editors, *Computer Aided Design of User Interfaces IV, CADUI' 2004*, pages 81–94, 2004.

[10] M.D. Harrison, M. Massink, and D. Latella. Engineering human flows in smart environments using formal techniques - full version. Technical Report 2009-TR-014, CNR-ISTI, 2009. http://puma.isti.cnr.it/ metadata.php?idcode=2009-TR-014&idcol=1&idauth=1&langver=it.

[11] J. Hillston. Fluid flow approximation of PEPA models. In *Proceedings of QEST'05*, pages 33–43. IEEE Computer Society, 2005.

[12] J.A. Hillston. *A compositional approach to performance modelling*. Cambridge University Press, 1996.

[13] G.J. Holzmann. *The SPIN Model Checker, Primer and Reference Manual*. Addison Wesley, 2003.

[14] C. Kray, G. Kortuem, and A. Krüger. Adaptive navigation support with public displays. In R. St. Amant, J. Riedl, and A. Jameson, editors, *Proceedings of IUI 2005*, pages 326–328, New York, 2005. ACM Press.

[15] M. Kwiatkowska, G. Norman, and D. Parker. PRISM: Probabilistic symbolic model checker. In T. Field, P.G. Harrison, J. Bradley, and U. Harder, editors, *Computer Performance Evaluation: Modelling Techniques and Tools Proceedings 12th International Conference, TOOLS 2002*, number 2324 in Springer Lecture Notes in Computer Science, page 200. Springer-Verlag, 2002.

[16] M. Kwiatkowska, G. Norman, and D. Parker. Analysis of a gossip protocol in PRISM. *ACM SIGMETRICS Performance Evaluation Review*, 36(3):17–22, 2008.

[17] T. McBryan and P. Gray. A model-based approach to supporting configuration in ubiquitous systems. In N. Graham and P. Palanque, editors, *Interactive systems: Design, Specification and Verification, DSVIS'08*, volume 5136 of *Springer Lecture Notes in Computer Science*, pages 167–180. Springer-Verlag, 2008.

[18] J McCarthy and P.C. Wright. *Technology as Experience*. MIT Press, 2004.

[19] R. Milner. *Communicating and mobile systems: the pi-calculus*. Cambridge University Press, 1999.

[20] Y. Rogers. Moving on from Weiser's vision of calm computing: Engaging UbiComp experiences. In Paul Dourish and Adrian Friday, editors, *Proceedings of Ubicomp 2006*, pages 404–421, Heidelberg, Berlin, New York, 2006. Springer.

[21] M.B. Rosson and J.M. Carroll. *Usability Engineering: scenario-based development of human computer interaction*. Morgan Kaufman, 2002.

[22] R. Ruksenas, J. Back, P. Curzon, and A. Blandford. Verification-guided modelling of salience and cognitive load. *Formal Aspects of Computing*, 2009. DOI: 10.1007/s00165-008-0102-7.

[23] J. Rushby. Using model checking to help discover mode confusions and other automation surprises. *Reliability Engineering and System Safety*, 75(2):167–177, February 2002.

[24] H.W. Thimbleby. Interaction walkthrough: evaluation of safety critical interactive systems. In G. Doherty and A. Blandford, editors, *Interactive Systems: Design, Specification and Verification*, number 4323 in Springer Lecture Notes in Computer Science, pages 52–66. Springer-Verlag, 2007.

[25] M. Tribastone. The PEPA plug-in project. In M. Harchol-Balter, M. Kwiatowska, and M. Telek, editors, *Proceedings of QEST'07*, pages 53–54. IEEE Computer Society Press, 2007.